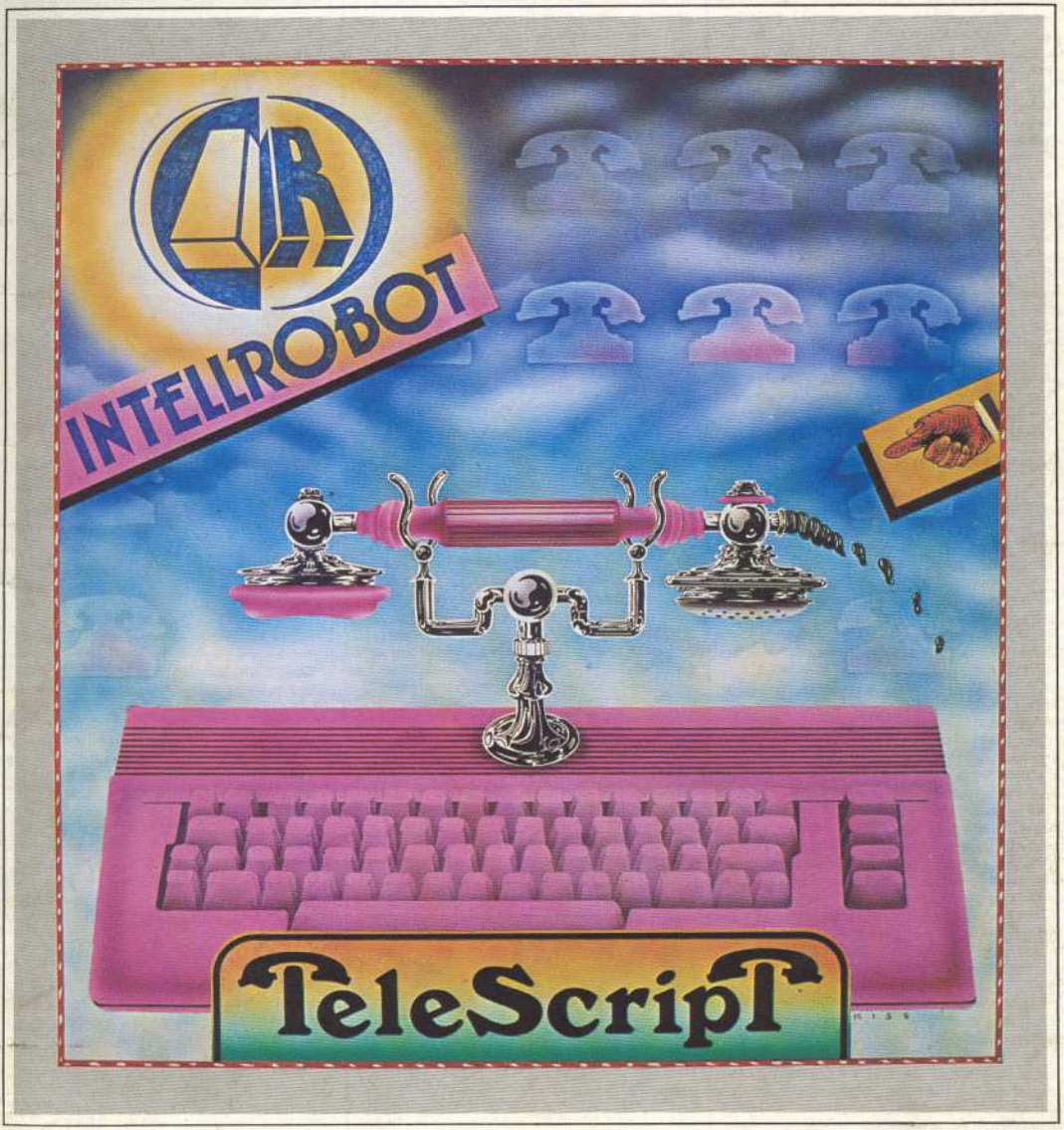
μ mikro számítógép magazin

Ára: 30 Ft



424-893 Pf. 156

1988/6

A SCANNTRONIK PALETTÁJA 1.

Szemet kapott a C64

Az NSZK-beli Scanntronik vállalat nem tartozik a nagy szoftverházakhoz, mégis a Commodore 64-es számítógépek tulajdonosait több hasznos programmal és hardverkiegészítéssel örvendeztette meg. Elsősorban grafikai és a Desktop Publishing (asztali kiadványszerkesztő) feladatok megoldását tűzte ki célul. Olyan programokat és kiegészítéseket kínál, amelyek ebben a gépkategóriában csaknem professzionális felhasználói igényeket elégítenek ki. Ilyen például a vonalas grafikai eredetiket letapogató és digitalizáló berendezése, a Superscanner II, valamint a hazánkban is — elterjedőben — lévő PRINTFOX nevű DTP program.

A következőkben rövid áttekintést adunk a Scanntronik termékekről, melyek főleg grafikus tervezőknek, újság-, illetve kiadványszerkesztőknek nyújtanak nagy segítséget munkájukhoz. De mindazoknak sikerélményt kínálnak, akik akár kedvtelésből, akár hivatásból szövegírással, szövegszerkesztéssel és grafikával foglalkoznak. A cég szoftverfejlesztéseit Hans Haberl irányítja, akinek neve fémjelzi a C64-re írt egyik legjobb rajzolóprogramot, a Hi Eddit is.

Superscanner II

Olvasóink közül bizonyára sokan vannak, akik már próbálkoztak botkormánnyal, egérrel, vagy más eszközzel komolyabb rajzot varázsolni a képernyőre. Nos, ha így volt, akkor tapasztalhatták, hogy ez nem is olyan egyszerű dolog. Ez a próbálkozás legtöbbször nem a kívánt eredményt adja, ráadásul elég „macerás" munka. Nagyon nehéz tehát ilyen módszerrel pontosan azt megrajzolni, amit elképzeltünk. Még nagyobb a feladat, ha kész tervet akarunk papírról átmásolni a képernyőre.

Ezeket a gondokat megoldja a Scanntronik képletapogató SUPERSCANNER II hardverkiegészítője. Ez lényegében egy gyufásdoboz nagyságú készülék, amelyet egy jobb minőségű (nem MPS típusú) mátrixnyomtató fejére lehet illeszteni. A kis letapogatóhoz adott szoftver biztosítja, hogy a nyomtatóba helyezett papírról a parányi optikai fej soronként letapogassa és a kazettacsatlakozó bemenetén keresztül a számítógépbe, illetve a képernyőre vigye át a vonalas fekete-fehér rajzot. A 640 × 400 pontból álló grafikai kép a képernyőn gördíthető és rendelkezésre áll még egy 320 × 200 pontos külön tároló is. A 80 × 50 mm-től a 200 × 280 mm-ig terjedő letapogatandó felületről ötféle felbontás érhető el, négyzetmilliméterenként maximálisan 72 pont sűrűségben. A monitoron kapott képet szükség szerint a szoftver saját rajzolóprogramjával tovább lehet módosítani, képpontonként javítani, lemezre menteni, illetve ki lehet nyomtatni. Az elmentett képfájl kompatibilis a többi Scanntronik programmal.

A külön kapható Scanner-Extension kiegészítés az Epson, a Star és a BX—100 nyomtató használatánál lehetővé teszi a még jobb (HQ) nyomtatási képet, de emellett tartalmaz még egy csokorra való hasznos felhasználói szoftverkiegészítést is. Mindenesetre a képletapogatósdi nem olcsó mulatság; a Superscanner II közel 400 DM-es ára meghaladja a C64-es alapgép árát is.

A Scanntronik egér és „sajtja"

A „nagy" számítógépek, mint a Macintoshok, az Atari ST-k és az Amigák működtetése egér nélkül nehezen képzelhető el. Sok C64-tulajdonos vágya, hogy gépéhez ilyen elektronikus egeret használhasson adatbeviteli eszközként. Az egér különösen rajzolásnál könnyíti meg a munkát. A Commodore 64-hez kétféle egér van forgalomban. Az egyik a botkormány-szimulátor egér, amely minden további nélkül használható a különböző programoknál. A kurzor, illetve az annak megfelelő jel, ceruza, ecset stb., nyolc irányba mozdítható a képernyőn. Gyorsabb és pontosabb vezérlés válik így lehetővé, de lendületes vonal mégsem húzható.

A másik egérfajta — és ilyen a SCANNTRONIC-MAUS IS — az úgynevezett valódi egér, melynek segítségével minden irányban rajzolhatunk a képernyőn. Ebben az esetben a monitoron a mozgás útja és annak sebessége pontosan megegyezik az egér mozgásával. A valódi egérnek két hátránya van. Az egyik, hogy csak a megfelelő szoftverrel installált programokkal hajlandó szóba állni, a másik, hogy a számítógép billentyűzete közben nem használható. A Scanntronik ezeket a feladatokat úgy oldotta meg, hogy kifejlesztette a MAUS•4 szoftverjét egy hozzá tartozó interfésszel. Négy programot tehetünk így alkalmassá arra, hogy valódi egeret használhassunk együtt a billentyűzettel.

Az NCE egérnek megfelelő Scanntronik egérhez tartozik egyébként a CHEESE-rajzoló-, illetve festőprogram is. Ez a Maus•4 nélkül is használható, mert teljesen menüvezérelt működésű és nem igényel billentyűhasználatot. A CHEESE —

Egy „valódi" C64 egér az interfésszel

A mátrixnyomtató képet „olvas"

A CHEESE festőprogram funkcióit a billentyűzet nélkül használhatjuk

ellentétben a Hi-Eddivel – nem nagyfelbontású grafikát használ, viszont pontosan ezért jobb a színezési lehetősége. A CHEESE ADD-ON az előbbi rajzolószoftverhez ajánlott kiegészítő programcsomag, amely a CHEESE képek Epson kompatibilis nyomtatókkal való kinyomtatását és a különböző programokkal való kapcsolatot biztosítják.

Segítségével a CHEESE-zel vagy más grafikai programmal előállított képeinket felhasználhatjuk saját BASIC-programjainkban. Sőt a képernyő tetszés szerint grafikai és szövegablakra bontható.

A C64-esek piacán egyedülálló a cég által kifejlesztett COLOURPRINTER szoftver, amely az egyes nyomtatókat színes mátrixnyomtatóvá változtatja, csupán a fekete nyomtatószalagot kell közben sárga, bíbor- és kék színű szalagra cserélgetni.

A Scanntronik Desktop Publishing programjairól a PRINTFOX-ról és a PAGEFOX-ról a következő számunkban írunk.

HUPPÁN BÉLA

μ mikro számítógép magazin

6. ÉVFOLYAM
1988/6. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Babos János (tervezőszerkesztő)
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Énekes Ferenc (KISZ)
Kovács Győző (levelezés)
Lindner László (sakk)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szebenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Varga András (TII, iskola-számítógép)
Vizessy Mária

Címképünk:
Kiss Ilona munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlap-előfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel u. 10/A)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86—0253

Szikra Lapnyomda
Budapest (88—0629)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Világútlevél	2
Feladatok-megoldások	11
Mit tud a C nyelv? II.	22
Merre tart a világ?	24
Közprogramok	29
Egy sarokkal olcsóbb!	30
Rendszerfejlesztési eszközök	30
μ '88. III. Országos Mikroszámítógépes Találkozó	32
μINFORM	37
Jobb verem Pascalban	38
Olvastunk . . .	39
Újföld	40
Adok-veszek-cserélek	44
Programtermék	46
ISKOLA-SZÁMÍTÓGÉP	
Az elektronizációs oktatási program eredményei és jövője	3
TechnoMIR	7
Nemes Tihamér Országos Középiskolai Számítástechnikai Tanulmányi Verseny	8
Zenei képességek vizsgálata számítógéppel	11
DIÁKROVAT	
Elrejtett változók	12
TOP-lista	12
ROM-lista Primóra	13
Segédvonal	13
Könyvtárprogram C64-re	14
Titkosírás	15
PROGRAMOZÁSTECHNIKA	
BASIC és gépi kód	16
A Precomp Plus előfordító program	18
Forgató szubrutinok 6502-re	20
Nyilvános adatbázisok	21
TERMÉKISMERTETŐ	
Az INTELL-GRAF szövegszerkesztő rendszer	35
μKLUB	
Áruk és áruk	36
SAKK	
Király plusz gyalog a király ellen III.	42
AZ OLVASÓ ÍRJA	45
KÖNYVEK	47
HÍREK, ÉRDEKESSÉGEK	47
PONTVADÁSZAT	48

Világútlevél

„A legszabadabb s legerősebb népek nem azok, amelyeknél egyes rétegek vagy kivételes egyes szellemek válnak ki műveltségük vagy más kiválóbb tulajdonságaik által, holott a nép zöme a sötétségben fetreng, hanem azok, amelyeknél a bár szerényebb fokú műveltség minél egyetemesebben hatja át az egészet."

(Herman Ottó: A háziipar és az iparművészet viszonya az anyagi gyarapodáshoz és a művelődéshez. Vasárnapi Újság, 1878. VI. 9. VI. 16.)

Állampolgári jogon ebben az évben először mentem át Hegyeshalomnál a határon egy rövid, egynapos megbeszélésre Bécsbe. Késő délután indultunk visszafelé, mellettünk egymás után húztak el H betűs kocsijaikon hazafelé igyekvő honfitársaink.

A határon beállva az egyik sorba — nyilván éppen oda, ahol a leglassabban kezelték az útleveleket (Murphy) —, elég időm volt arra, hogy megfigyeljem ki, mit hozott haza a „bécsi út másik végén" tett portyájáról. Nem csekély örömömre szolgált, hogy a behozott áruk java része még mindig számítógép és különféle periféria volt. A számítógépekhez viszonyítva hozzávetőlegesen negyedannyian hoztak videomagnót és rádiót. Úgy látszott, hogy a legkevesebben háztartási gépeket vásároltak, valószínűleg azért, mert ezek nagy része nem érte el a vámmentesen behozható érték felső határát, és így nekik nem kellett a vámpapírokért sorba állni.

Kihasználva a várakozás nem túl lélekemelő perceit, igyekeztem kilesni, hogy mit is rejtenek a számítógépes csomagok, ki milyen eszközt hoz be saját gyönyörűségére, vagy — valószínűleg nem kevesen — eladásra. Már első látásra is szembetűnő volt, hogy a Commodore és Sinclair szerelem továbbra is tartósnak látszik, a legtöbben ezeket a gépeket vásárolták. Volt azonban — nem is kevés — olyan gép is, amilyenről még csak nem is hallottam. Amikor az egyik várakozót megkérdeztem, hogy miért éppen ezt az ismeretlen gépet vette meg, elmondta, hogy egyrészt azért, mert az egyik Mariahilferstrasse-i ügyes — „magyarul is beszélünk" — kereskedő olcsón adta, másrészt az eladó azt is mondta, hogy a gépet nagyon kis befektetéssel Spectrum-kompatibilissé lehet tenni.

Nem akartam a szemlátomást nagyon boldog tulajdonost elkeseríteni, de ezekben a „majdnem kompatibilis" ügyekben, főleg olvasóink leveleit böngészve, egyre kevésbé hiszek. A µM szerkesztőségének sokszor írnak lelkes, de gyakran nagyon kiábrándult amatőrök, hogy a viszonylag olcsón vett géphez kaptak ugyan egy BASIC fordítót meg egy-két játékkazettát, amit persze nagyon gyorsan meguntak, de a kereskedő azt az igéretét, hogy később lesz majd más is, például szövegszerkesztő, számolótábla, adatbázis-kezelő vagy esetleg ügyes rajzoló- és szerkesztőprogram, később nem teljesítette. Ez persze nemcsak Bécsben van így, de még itthon is, tudniillik egy sokat tudó szoftver fejlesztési költsége meglehetősen magas, és ha nincs arra remény, hogy a termékből néhány ezer eladható, akkor senki sem kockáztatja meg a befektetést. Néhány száz eladott gép esetében így érthető, ha a kereskedő csak ígér, de majdnem biztos, hogy sohasem szállít. Ráadásul a vevő — ez a tapasztalatom — a hardver bűvöletében él, kevesen tudnak egy tetszetősen tervezett és csomagolt, szép nevű és főleg olcsó gépnek ellenállni.

Ez a hardverszemlélet persze nemcsak az amatőrökre jellemző, így vannak ezzel sokszor még az állami intézmények is, amikor felvásárolják a házi számítógépeket, hogy azokra szervezzék meg egy nagyvállalat teljes információs rendszerét.

Sok iskolát is láttam, ahol ott vannak a számítógépek, de megfelelő szoftverkörnyezet hiányában alig használják a gépeket, mert nincs elég, tartalmilag is és pedagógiailag is jól megszerkesztett oktatóprogram. Ezért marad a BASIC programozás tanítása, ahelyett, hogy a különböző tantárgyak tanulására alkalmaznák a gépeket.

Így azután az a helytelen kép alakul ki az iskolákban és a közvéleményben is, hogy a számítástechnika egyenlő a programozással, amit persze a nem számítástechnikus felhasználó nem akar megtanulni. Pedig ha lenne elegendő okos és főleg felhasználóközeli szoftver, akkor a sok pénzért megvásárolt otthoni számítógépet sem csak a programozói ambicióktól fűtött gyerekek használnák, hanem az egész család, mindenki arra, amire a napi munkája során szüksége lehet.

Számomra azért volt meghökkentő a határon látott számítógépes felvonulás, mert ezeknek a gépeknek a nagy részét valószínűleg úgy vették meg, hogy a majdani alkalmazásra nem is gondoltak a vásárlók. A már említett ismeretlen honfitársammal való beszélgetés során is kiderült, hogy az üzletben szóba sem jött, hogy milyen a gép szoftverellátottsága, vannak-e hozzá programcsomagok (... majd a gyerek megszerzi, ha kell neki — mondta az apuka).

Azt hiszem, arra sem sokan gondolnak, hogy milyen előnyökkel jár, ha olyan gépet vesznek meg, amit itthon is árulnak, vagy amiből már nagyon sokat hoztak be az országba. Ha ilyen gépet vesznek, akkor legalább arra van lehetőségük, hogy eljárjanak a különféle számítógépes klubokba, például az NJSZT HCC valamelyik szekciójába, és ott programokat cseréljenek, különféle hasznos fogásokat tanuljanak.

Figyelve a vámosok munkáját, feltűnik, hogy a hazai számítástechnikai kultúra szempontjából igazán értékes PC-kompatibilis gépeket (mi is elkezdhetnénk ezt a rövidítést használni) nem láttam. Valószínűleg azért nem, mert ma sajnos semmi sem ösztönzi a vásárlókat arra, hogy ne játék-, hanem professzionális berendezésekre fordítsák a pénzüket. Aki magának hoz egy ilyen gépet, az a behozatali eljárás bonyolultságára panaszkodik, aki meg eladásra, azt a sokat bírált, hírhedt 25 ezres rendelet sújtja: ezeket a gépeket ugyanis, miután az értékük a 25 ezer forintot meghaladja, közületek nem vásárolhatják meg. Rövid „határszemlém" után bizton állíthatom, hogy a rendelet elérte a célját: igazán használható, azaz PC-kompatibilis számítógép a vámolásra váró utasok kezében egy sem volt. Más kérdés persze, hogy érdemes-e örülnünk ennek az „eredménynek". Számomra ugyanis egyáltalán nem kétséges, hogy a vállalatok továbbra is vásárolnak magán-importból származó gépeket, persze ismét visszatérnek a C64 és C128-as készülékekhez, ahelyett, hogy PC-kompatibilis gépekből elégítenék ki az igényeiket. Így ismét elkezdődik a programfejlesztés ezekre a professzionális célra alkalmatlan gépekre. Hiszen házi számítógépekről lévén szó, megfelelő kész és főleg használható vállalati célú programeszközöket vagy programcsomagokat beszerezni ezután sem lehet. Ráadásul a felhasználók által kifejlesztett programcsomagokat legfeljebb majd az országhatáron belül lehet máshol is felhasználni. Ezeket a sok pénzbe került programokat nem lehet exportálni sem, hiszen se a tőkés, se a szocialista országokban házi számítógépeket a termelésben már nem használnak.

Van azonban ennek a jelenlegi magánimport-orientációnak egy másik, még roszszabb irányba mutató hatása is. Nem hivatalos becslések szerint ma már több tízezer házi számítógép van a lakásokban. Ha jól meggondoljuk, ezeknek a gépeknek az összteljesítménye lassan összemérhető lesz a termelésben használt személyi számítógépek teljesítményével. Márpedig ha ez így van, akkor látszik csak igazán, hogy milyen haszna lett volna annak, ha a számítógépek magánúton való beszerzését okos, gazdasági szabályozókkal úgy irányítjuk, hogy magánhasználatra is inkább professzionális gépeket vásároljanak, mint házi számítógépeket.

Beláthatatlan volna annak a haszna, ha az importált gépek jó része PC-kompatibilis lenne, és így kialakulhatnának az otthoni munkahelyek, értelmes munkát adva például gyes vagy gyed mellett a mamáknak, nem is beszélve az egészségükben károsultakról, kiváltképp a mozgássérültekről és a vak, illetve a gyengénlátó emberekről, akiknek sokszor megoldhatatlan problémát jelent a közlekedés.

Valószínűnek tartom, hogy hasonló meggondolások alapján határozták el a szomszédos Csehszlovákiában, hogy a társadalom számára annyira fontos számítógépek magánimportját felszabadítják és ezekre a berendezésekre eltörlik a vámfizetési kötelezettséget. Már nekünk is ezt kellett volna tennünk nagyon régen. Akkor ugyanis a vásárlásnál és a behozatalnál a fő rendező elv valószínűleg a használhatóság és nem az alacsony vámösszeg lett volna. Ennek nyilvánvalóan az volna az eredménye, hogy a vásárlók nagy része inkább PC-kompatibilis gépeket vásárolt volna magán- és eladási célokra egyaránt. Biztos vagyok benne, hogy ezután is lennének „nyerészkedők", de nem hiszem, hogy többen, mint ahányan ma ugyanezt a 25 ezer forintnál olcsóbb gépekkel teszik.

Ami az eredményt illeti, nem lehet kétséges, hogy a számítástechnika társadalmi méretű elterjedése szempontjából, de az ország gazdasági érdekeit tekintve is egy sokkal liberálisabb, a nagyobb gépek magánimportját elősegítő pénzügyi és vámpolitikára lenne szükség, hogy az egyre kevésbé használható gépek beáramlását megszüntessük.

KOVÁCS GYŐZŐ

Az elektronizációs oktatási program eredményei, gondjai és jövője

A közelmúltban részletes elemzés készült a program előrehaladásáról, eredményeiről, gondjairól, és miután igen széles körben érdeklődésre számot tartó programról van szó, úgy gondolom, hogy indokolt a nagy nyilvánosság elé tárni értékelésünket, a jövőre vonatkozó elképzeléseinket, problémáinkat, eredményeinket.

Mi volt a program célja?

A számítástechnika-oktatási program és az annak keretében elindított iskolaszámítógép-program szerteágazó célokat tűzött maga elé, mind a tartalmi kérdéseket, mind az oktatás fokozatait illetően. Az eredeti, 1968-ban jóváhagyott program az oktatás alábbi három körét különböztette meg:

a) a számítástechnikai alapismeretek oktatása
b) a számítástechnikai alkalmazási ismeretek oktatása
c) a számítástechnikai szakemberképzés.

A számítástechnikai **alapismeretek oktatása** azokat az általános ismereteket szolgálja, amelyek egyrészt a számítástechnikai kultúra széles körű elterjesztését megalapozzák, másrészt amelyekre az alkalmazási ismeretek oktatása, illetve a szakemberképzés épülhet. A számítástechnikai **szakemberképzés** számítástechnikai eszközök és programok (hardver és szoftver) fejlesztésére és előállítására, üzemeltetésére, karbantartási feladatok ellátására képez ki szakembereket. A számítástechnikai **alkalmazási ismeretek oktatása** pedig arra készít fel, hogy a számítástechnikát a népgazdaság meghatározott területén hatékonyan és magas szinten tudják **alkalmazni** a nem számítástechnikai szakemberek (szakmunkások, technikusok, mérnökök, jogászok stb.).

Az elektronizáció gazdaságfejlesztési programjának létrejöttével kibővült, átalakult a korábbi számítástechnikai oktatási program.

Az elképzelések szerint az új programnak az oktatás, képzés minél szélesebb területét kell figyelembe vennie, mert várhatóan a társadalom egésze kapcsolatba kerül az elektronikával, informatikával. Az informatika oktatását ezért indokolt az egész oktatási, képzési rendszerre kiterjeszteni, hogy

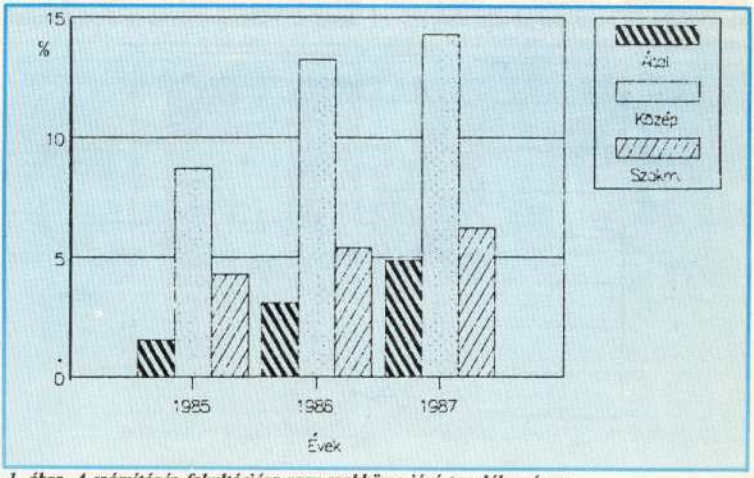
- az informatika alapjaival megismerkedjék az egész társadalom;
- az informatika-elektronika alkalmazására felkészüljenek mindazok, akik azt munkájukban használhatják;
- az informatika-elektronika szakterületére szakemberek képeztessenek ki.

Ezekre a meggondolásokra alapozva a fejlesztés első tézisét abban fogalmaztuk meg, hogy **az oktatás-képzés teljes keresztmetszetére és minden szintjére át kell gondolni és részleteiben kidolgozni egy rendszerszemléletű oktatási tervet,** amelyben az egyes képzési, oktatási követelmények egymásra épülve, a kölcsönhatásokat szem előtt tartva, a párhuzamosságokat kiiktatva, a hiányokat megszüntetve minden elem a rendszer lehető legnagyobb hatékonyságát szolgálja.

A fejlesztés második tézise az volt, hogy az oktatás tartalmának meghatározásában ami a tananyag teljes megismerése mellett az egyes szaktantárgyakban a részletek feltárását követeli meg.

A fejlesztés harmadik tézise pedig az volt, hogy **az informatika oktatási rendszere hierarchikus felépítésű legyen.** Ez azt a követelményt fogalmazza meg, hogy az egyes szinteken oktatott ismeretek épüljenek az alsóbb szinten oktatottakra, de nyújtsanak önmagukban is befejezett oktatást, illetve képzést.

A program megkülönböztetett figyelmet fordított az új tanárok számítástechnikai ismeretekkel való felvértezésére és az iskolákban tanító tanárok továbbképzésére. Meghatározta azokat a feltételeket, amelyeket a program megvalósításához biztosítani kell, így az iskolák megfelelő számú és korszerű



1. ábra. A számítógép-fakultációra vagy szakkörre járó tanulók aránya

érvényesüljön és **hasznosuljon az informatikának az az előnye, hogy az egyes szakterületek ismeretanyagának egy, a korábbinál magasabb szintű szintetizálását és integrálását teszi lehetővé** (automatizált műszaki tervezés, gyártás, mechatronika stb.).

Az ismeretek ilyen rendszerezése megkönnyíti a szakismeretek körében a tények, módszerek, eredmények megértését, a képesség kifejlesztését ezek közvetlen alkalmazására, segítséget nyújt arra, hogy hivatkozásért, tanácsért hová lehet fordulni, és érzékelteti az ismeretanyaghoz kapcsolódó szakterület gazdasági vetületét. Az informatika tehát az oktatás eszközeinek bővítésével didaktikai és technikai segítséget nyújt. Ezek az informatika nyújtotta lehetőségek már a jelenlegi tapasztalatok alapján is azt a kényszerítő erőt jelentik a tanulásban, eszközökkel való felszerelését, oktató programcsomagok, tananyagok, könyvek megjelentetését stb.

Eredmények

Az elmúlt 5 évben jelentős előrehaladás történt az eszközellátásban, évről évre folyamatosan növekedett az elektronikai-informatikai oktatásban részt vevők száma *(1. ábra).* Megjelentek azok az új, korszerű eszközök, amelyek lehetővé teszik a számítógépeknek a gépekkel, műszerekkel való összekapcsolását, azok vezérlését, mérési adatok gyűjtését és feldolgozását. A szakoktatásban is jelentős előrelépés történt a számjegyvezérléses gépek, számítógéppel segített tervezési eszközök, robotok, korsze-

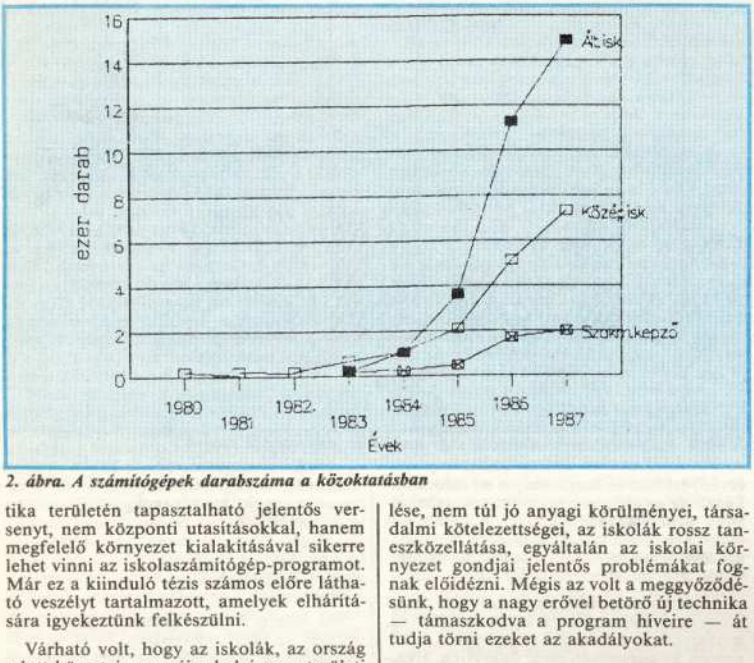
rű információs feldolgozási módszerek elterjesztésében. Több száz, különféle gépre alkalmazható programokra bővült az oktató programcsomagok választéka, számtalan könyv jelent meg, igen sok tanfolyamot szerveztek különféle intézmények, úgynevezett „C" szakos számítástechnikai tanárképzés indult meg, bővült a technika, számítástechnika szakos tanárok száma. Az eredmények alapján azt lehetne gondolni, hogy a program minden szempontból a kívánatos módon halad előre és nincs más teendő, mint a megkezdett úton továbbhaladni.

Gondok a program megvalósításában

A látványos eredmények mellett természetesen vannak jelentős problémák, sőt kudarcok is. A program indításakor azzal a feltételezéssel éltünk és úgy gondoljuk, ez a feltételezés ma is helytálló, hogy újszerű módszerek alkalmazásával, nem kötelező előírásokkal, az iskolai és tanári kezdeményezőkészségre támaszkodva, kihasználva az országban a számítástechnika, informatika területén tapasztalható jelentős versenyt, nem központi utasításokkal, hanem megfelelő környezet kialakításával sikerre lehet vinni az iskolaszámítógép-programot. Már ez a kiinduló tézis számos előre látható veszélyt tartalmazott, amelyek elhárítására igyekeztünk felkészülni.

Várható volt, hogy az iskolák, az ország adott körzetei, megyéi a helyi vagy területi adottságoktól függően eltérő módon fogják értelmezni a feladatokat, és ennek megfelelően különféle módon fogják az új ismereteket az iskolákba bevezetni, eltérő feltételeket fognak biztosítani, ami már önmagában is egy differenciált megoldást jelent. De emellett várható volt, hogy a program iránt elkötelezett diákok, tanárok, iskolák, megyei irányító szervek jelentős mértékben kívánják érvényesíteni saját elképzeléseiket, amelyek nem mindig fognak egybeesni a központi elképzelésekkel. Mégis a program tervezői, irányítói úgy gondolták, hogy éppen ez a sokszínűség fog hozzájárulni a megfelelő módszerek kialakításához, elterjedéséhez és ez a sokszínűség fog lehetőséget adni arra, hogy mind a diákok, mind a tanárok, mind az iskolák közül a kiemelkedő teljesítményt nyújtók jelentős mértékben hozzájárulnak majd a program sikeréhez.

A számítástechnikában tapasztalható versenytől azt vártuk, hogy a számítástechnikai eszközöket, gépeket, programokat előállítók versenye egyrészt letöri a magas árakat, másrészt magasabb színvonalú eszközöket fog a versenyen keresztül biztosítani. Számítottunk arra is, hogy az iskolákat fenntartó közösségek és környezetük anyagi és szellemi erőforrásaikkal is hozzá fognak járulni az előrehaladáshoz. De számítottunk arra is, hogy az iskolákban és a társadalomban jelen lévő konzervativizmus, a rossz beidegződések, az új technikáktól és az új feladatoktól való félelem, a klasszikus kultúra féltése, az iskolák és környezetük eltérő anyagi viszonyai, egyáltalán eltérő társadalmi környezetük, a tanárok túlterhelése, nem túl jó anyagi körülményei, társadalmi kötelezettségei, az iskolák rossz taneszközellátása, egyáltalán az iskolai környezet gondjai jelentős problémákat fognak előidézni. Mégis az volt a meggyőződésünk, hogy a nagy erővel betörő új technika — támaszkodva a program híveire — át tudja törni ezeket az akadályokat.



2. ábra. A számítógépek darabszáma a közoktatásban

A mai helyzet

A várakozásoknak megfelelően rohamosan nőtt a számítástechnikai eszközök száma az iskolákban *(2. ábra)*. Különösen feltűnő a növekedés az általános iskolákban, ahol ma már a kezdeti, 1985—86-os tanév elejei 25%-kal szemben az iskolák több mint 75%-ában van számítógép *(3. ábra)*. Jelentősen nőtt az egy iskolára jutó számítógépek száma és egy számítógépre egyre kevesebb tanuló jut *(4. ábra)*.

Ugyanakkor a hazai számítógépgyártás és -értékesítés ellentmondásai miatt csak részben sikerült egységes gépparkot kialakítani, bár a tanév végén nyilvántartott gépparkon belül tömegét és arányait tekintve elsősorban a Commodore cég (12 588), a Videoton (6785) és a Híradástechnika Szövetkezet (2423) gépei adják a túlnyomó többséget. Sajnálatos módon jelentős számban vannak más géptípusok is.

A **számítógéppark** sokszínűségének kialakulásában a következő tényezők játszottak alapvető szerepet. Nem volt módunk arra, hogy egy időben vásároljuk meg az öt évre tervezett gépmennyiséget, ugyanakkor a hazai gyártók sem tudtak volna vállalkozni ötéves gépigény kielégítésére import- és termelési kapacitás hiánya miatt, és nem volt mód arra sem — devizahiány miatt —, hogy külföldről szerezzük be a tervezett mennyiséget. Ismerve a számítógépipar dinamikus fejlődését, ezek után természetes következmény volt, hogy évente, kétévente a gyártócégek új géptípusokkal jelentek meg. Ennek megfelelően eleve többféle gépet lehetett és kellett beszerezni. Az egész iskolaszámítógép-program ideje alatt, de az országos számítástechnikai központi fejlesztési program ideje alatt is országos probléma volt és ma is az, hogy az előbb említett okok miatt nem lehetett ez ideig az országban sem egységes gépparkot kialakítani. Ténykérdés azonban az is, hogy a világ összes fejlett országában sokszínű géppark működik és ehhez alkalmazkodik az oktatás is.

Állandóan visszatérő probléma volt, hogy az iskolákat hazai gyártású vagy külföldről beszerzett gépekkel lássuk-e el. Az eddigi tapasztalat azt mutatja, hogy a hazai ipar minden gondja, problémája mellett is még mindig lényegesen megbízhatóbb partner, mint az ország devizahelyzetétől függő, különféle érdekű importáló vállalatok, amelyek rendeléseik teljesítésénél teljes mértékben az importengedélyezés mechanizmusától függően és a szigorú nyereségérdekeltségű gazdálkodási szabályok figyelembevételével gazdálkodnak. Az is kétségtelen tény, hogy a hazai gyártásnál is megjelenik az importfüggés, de ennek mértéke az elmúlt időszakban jelentősen csökkent az importhányad folyamatos csökkenése következtében.

A számítógép-importálás gondjait nagyon jól tükrözi a C16-os számítógép esete: a II. iskolaszámítógép-pályázaton első helyezést ért el; nem egészen negyedévvel a pályázat eredményének kihirdetése után gyártását megszüntették és helyette megjelent a C+4-es gép. Ehhez egy évig lehetett hozzájutni, majd itt is zavarok keletkeztek a szállításban. Az egyik forgalmazó cég például 1986 szeptemberében ajánlatot tett 1986. év végi szállítással közel kétezer C+4 behozatalára, de ezek a gépek a mai napig nem érkeztek meg. Ugyanakkor egy másik kereskedelmi tranzakció keretében behoztak több mint 10 ezer Enterprise gépet; ára lényegesen magasabb, mint a C+4-es gépé, és azzal nem kompatibilis, ezenkívül

gyártása megszűnt és a perifériák sincsenek biztosítva hozzá. A kialakult helyzetben ezért célszerű a Magyarországon előállított, Videoton gyártmányú gépek beszerzése, amelyek megfelelnek hosszabb távon is az iskolai követelményeknek és jó a periféria-választékuk is.

A gépek üzemeltetésével összefüggésben megjelentek a **szervizelés problémái** is. Minden szállító a hazai előírások szerint vállalkozott a garanciális és garanciaidőn túli javítások megszervezésére. A javítási költségek azonban elég magasak, különösen a tőkés importból beszerzett gépek esetén, mivel a javítást végző szervizek a hazai árképzési elveknek megfelelően jelentős összegeket kérnek javításért. Az átalánydíjas javításokra vonatkozó szerződéseiket viszont az iskolák általában nem vették igénybe, véleményünk szerint rosszul értelmezett takarékosságból. De tapasztaljuk azt is igen sok esetben, hogy az iskolák panaszkodnak a gépek hibáira, de minden felkérés, ösztönzés ellenére sem keresik meg a szervizeket.

Céljainknak megfelelően jelentős számban vásároltak az iskolák korszerű perifériákat: nyomtatókat, hajlékony mágneslemez-egységeket, különféle interfészeket. Megjelentek az iskolákban a számjegyvezérléses szerszámgépek, elektronikus mérőeszközök, robotok, automatizált műszaki tervezést bemutató rendszerek.

Az 1987/88. tanév elején már több mint 1200 nyomtató, 1400 hajlékony mágneslemez-meghajtó és több mint 50 számjegyvezérléses szerszámgép, valamint jelentős számú egyéb periféria, interfész működött.

A legalapvetőbbnek mégiscsak azt kell tekintenünk, hogy a TII, mint nagy vásárló, az összes probléma és gond mellett is letörte a magas árakat, és ma lényegesen — 4-5-ször — alacsonyabb árakon korszerűbb számítógépekhez lehet hozzájutni, mint 1983—84-ben.

A központi források beszűkülése és ennek alapján az Állami Tervbizottság határozata arra késztette a program irányítóit, hogy a számítástechnikai **program** megvalósításának **finanszírozásába** vonja be az iskolákat fenntartó tanácsokat, megyei tanácsokat és vegye igénybe a szülők, vállalatok anyagi támogatását is. Ennek mechanizmusát a Tudományszervezési és Informatikai Intézetnek az egyes megyékkel megkötött szerződései szabályozzák. Tapasztalataink szerint ez a megállapodás jól működik, bár a dolgok természetéből eredően időszakosan konfliktusos helyzet is kialakul, hiszen a TII arra törekszik, hogy az országban a lehetőségekhez képest egységes géppark legyen, ugyanakkor az egyes városokban, megyékben jelen vannak a hazai számítógépgyártók és -importálók, akik természetesen el akarják adni eszközeiket az iskoláknak. Sok esetben ezek az eszközök a beszerzési forrásuk miatt nem rendelkeznek garanciával, magas az áruk és nincs megoldva a javítószolgálatuk, szervizhátterük. Ilyen esetekben nem lehet támogatást nyújtani. Más esetekben viszont ezeket az eszközbeszerzési forrásokat — ha megfelelnek az általános követelményeknek — bevontuk a gépbeszerzések körébe. Egyértelműen leszögezhető, hogy nem a gyártók és a szállítók vannak kedvezményezett helyzetben, hanem meghatározott géptípusok, függetlenül azok szállítóitól.

A **szoftverellátás** még színesebb képet mutat, mint az eszközellátás. Az alapvető problémát itt az okozza, hogy Magyarországon is — ahogy más országokban, a fejlett ipari országokban is — mind a mai napig még keresik azokat a területeit az oktatásnak, ahol a számítástechnikát, informatikát, elektronikát a leghatékonyabban lehet hasznosítani. A legtöbb országban vita folyik arról, hogy a korszerű ismeretek átadására mi a legkedvezőbb forma: az általában elterjedt kötetlen szabadfoglalkozások rendszere vagy a néhány országban elképzelt tantárgyi forma. Egyáltalán az a vitakérdés, hogy egységesen kell-e számítástechnikát, informatikát tanítani mindenkinek, vagy céloktól függően, igen differenciált és tagolt módon — ahogyan ezt Magyarországon is gondoljuk — kell ezt az oktatást megvalósítani.

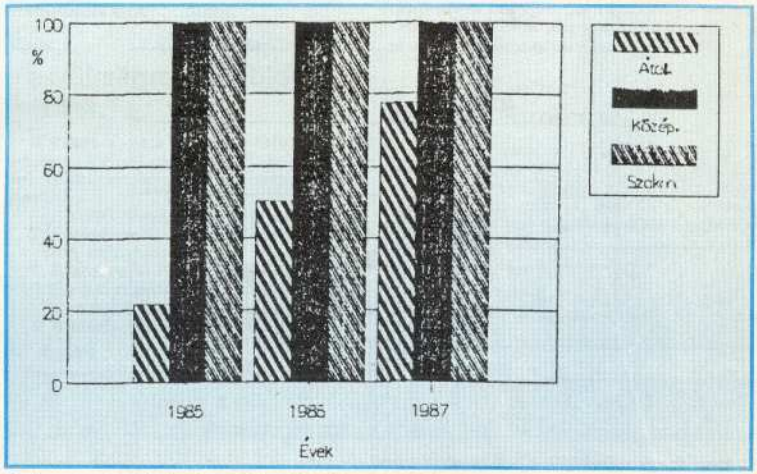
Mi úgy gondoljuk, hogy ez a vita természetes, hiszen több száz éve tanított tantárgyak belső tartalmát, oktatási módszereit is időről időre a követelményeknek megfelelően módosítani kell. Gondoljunk itt az új matematikára, a természettudományi tantárgyak korszerűsítésére vagy a magyar irodalom, a nyelvek oktatásának problémáira. Véleményünk szerint ma már nem szabad olyan kötelező állásfoglalást kialakítani, amely bármely elképzelést vagy módszert egyedül üdvözítővé tenne, hanem arra kell módot adni az iskoláknak, hogy adott lehetőségeiket, az iskola oktatási céljait figyelembe véve maguk döntsék el, milyen módszereket alkalmaznak.

Nyilvánvaló, hogy az informatikai szakemberképző iskolákban tantárgyi keretekben és a klasszikus oktatás módszereit felhasználva kell az oktatást megszervezni, de itt is teret kell adni a rugalmas oktatási formáknak. Már egészen más a helyzet azokban a szakmai képzést végző iskolákban, ahol az informatikát, elektronikát alkalmazzák, de nem cél az informatikai szakemberképzés. Itt az a feladat, hogy a leendő szakmunkásokat, technikusokat felkészítsék az informatika, elektronika saját szakmájukban való hatékony alkalmazására. Természetesen itt is elképzelhető kötelező tantárgyként, de a tapasztalatok szerint lényegesen hatékonyabb, ha ezeket az ismereteket beleépítik szakmai tantárgyaikba és alkalmazzák a rugalmas oktatási formákat.

Megint más a helyzet a gimnáziumokban, ahol a számítástechnikát, informatikát, elektronikát az általános műveltség részeként oktatják és emellett adnak konkrét alkalmazási ismereteket. Itt minden bizonnyal célszerű élni a fakultáció lehetőségével, és különösen fontosnak ítélhetők a szakköri és egyéb foglalkozási formák. Nyilvánvalóan egészen más követelményeket kell támasztani az általános iskolákban, ahol az alapvető ismeretekkel kell megismerkedni.

Éppen ez a sokszínűség bizonyítja, hogy még hosszú időn keresztül szükséges az iskolai gyakorlat figyelemmel kísérése, pedagógiai, módszertani anyagok összegyűjtése, ezek tudományos elemzése, és megengedhetetlen bármilyen voluntarista, a tényeket figyelmen kívül hagyó, erőszakos megoldás alkalmazása, amely minden bizonnyal magában hordozná a későbbi problémákat.

E bonyolult helyzet miatt a számítógépes oktatási programok kidolgozása igen nagy körültekintést igényel. Nagy eredménynek tartjuk, hogy ilyen nehéz helyzetben ma már több száz különféle géphez és tantárgyhoz használható program áll rendelkezésre, ugyanakkor óriási problémának tartjuk, hogy ezen a területen nem tudunk elég gyorsan előrehaladni, elsősorban anyagi források hiánya miatt. Ezért erőfeszítéseket teszünk, hogy e célra a mainál lényegesen nagyobb források álljanak az ok-



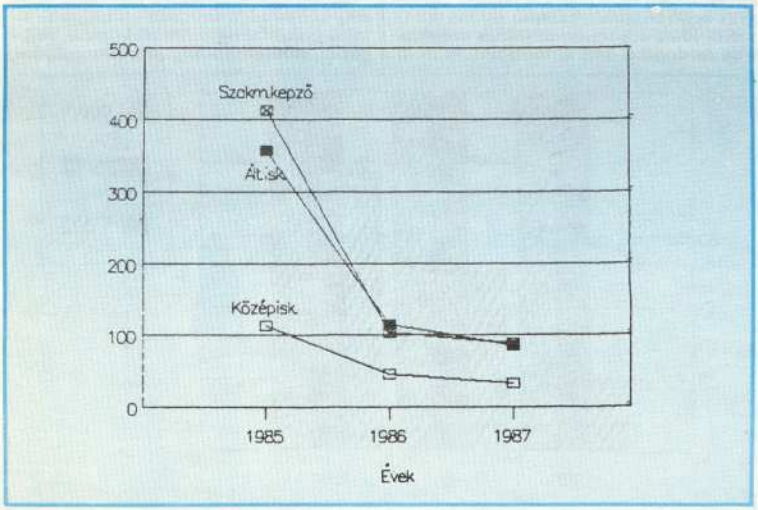
3. ábra. A számítógéppel rendelkező iskolák száma (%)

tatás rendelkezésére, hiszen ilyen források nélkül nincs mód olyan teamek létrehozására, ahol pedagógus, informatikus együtt tudna dolgozni.

Világosan látnunk kell, hogy a szoftverfejlesztésben való minőségi és mennyiségi előrelépés olyan követelményeket támaszt a pedagógusokkal szemben, amelyeknek idő, anyagi és erkölcsi elismerés hiányában nem tudnak eleget tenni vagy csak a pedagógusok egy igen kis köre tud erre vállalkozni. Mindenképpen meg kell keresnünk azokat a módszereket, lehetőségeket, amelyekkel ezt a kört bővíteni tudjuk. Úgy gondoljuk, hogy van egy tisztességes határa annak, hogy milyen mértékben vesszük igénybe a pedagógusok áldozatkészségét. Ha nem tudjuk honorálni valamilyen módon ezt az erőfeszítést, akkor ez előbb-utóbb visszaüt és súlyos gátjává válik az előrehaladásnak. Ennek jeleit egyre inkább tapasztalhatjuk, főleg az egyre nehezülő gazdasági körülmények között.

Az iskolaszámítógép-program kulcsfigurája a tanár. A program megindításától kezdve visszatérő gond volt, mennyiben leképző főiskolákon és mostantól kezdve a tanítóképző főiskolákon is. A tanárképzésben a legnagyobb gondot nem a számítástechnikai ismeretek oktatása okozta, hanem mindenekelőtt azok a pedagógiai módszerek, amelyeknek alkalmazására a leendő tanárokat meg kell tanítani. Ez, mint jeleztük, egy hosszabb folyamat, ezért már hosszabb ideje külön kutató-elemző munka folyik az iskolákban alkalmazható pedagógiai módszerek összegyűjtésére, a külföldi tapasztalatok hasznosítására. Amilyen mértékben halad előre a program az iskolákban, olyan mértékben sikerül majd mind több és több tapasztalatot összegyűjteni, és így folyamatosan egyre korszerűbbé válik a számítástechnikához értő tanárok képzése is. E folyamatban jelentős szerepet játszik a technika szakos, valamint a diploma utáni, úgynevezett „C" szakos tanárképzés. Ennek mértékét jelenleg azonban anyagi források hiánya korlátozza. Az egyik legsürgetőbb feladat, hogy ezen mielőbb változtassunk.

A program indításakor nyilvánvaló volt, hogy az iskolákba kerülő gépek hatékony



4. ábra. Az egy számítógépre jutó tanulók száma

het hasznosítani az iskolában a matematika, fizika szakos tanároknak a közel 20 éve tanított „nagygépes" ismereteit, a programnyelveket, illetve milyen ütemben lehet felkészíteni a leendő tanárokat a korszerű személyi számítógépek használatára. Még 1981-ben — amikor megkezdődött az iskolaszámítógép-program kimunkálása, tehát még a program elindítása előtt — az Eötvös Loránd Tudományegyetemen megkezdődött a felkészülés a korszerű személyi számítógépek használatára. Így a program indításakor már néhány száz személyiszámítógép-ismerettel rendelkező tanár csatlakozhatott a több mint 15 ezer „nagyszámítógépes" gyakorlattal rendelkező tanárhoz.

Az elmúlt évek során folyamatosan bővültek a korszerű számítógépes oktatás feltételei a tudományegyetemeken, a tanárkihasználásához **szükség van az iskolákban működő tanárok továbbképzésére is.** Ezt a célt szolgálta a gépek átadásakor előírt rövid oktatás és a tanárok számára szervezett különféle továbbképzések is. Teljesen nyilvánvaló, hogy a rövid tanfolyamok csak egy minimális ismeretanyagot, „induló tőkét" adhatnak a tanároknak, és szükség van lényegesen nagyobb szintű, szerteágazó továbbtanulási lehetőség biztosítására. Az elmúlt időszak tapasztalatai alapján a legsürgetőbb feladat a továbbképzés új módszereinek megtalálása.

Megítélésünk szerint a tanárok számára olyan modulrendszerű továbbképzésre van szükség, amely lehetővé teszi az új ismeretek megszerzését a legkülönfélébb területeken. Kiemelt fontosságúnak tartjuk az egyes szaktantárgyak oktatásával összefüggő pedagógiai-módszertani továbbképzést, emellett szükséges olyan tanfolyamok szervezése, amelyeken a különféle alkalmazásokat ismerik meg a tanárok. Egy másik tanfolyamon a különféle programnyelvek vagy magas színvonalú, korszerű számítógépes szolgáltatások elsajátítása lehetne a cél (adatbázis-kezelés, szövegszerkesztés, grafika); további lehetőség lenne szoftverírás, oktatóprogram-csomag tervezése és használata.

Örömmel üdvözöljük másrészt, hogy az új oktatási törvény egyre nagyobb szabadságot ad az iskoláknak és a pedagógusoknak alkotó munkájuk kibontakoztatásához. Az iskolák önállóságának növekedése — ellentmondásos módon — ugyanakkor gátjává válhat a program gyors ütemű megvalósításának, mert az egyes iskolák konkrét érdekei, belső erőviszonyai, az iskola vezetésének esetleges konzervativizmusa, a tanári kar belső gazdasági érdekellentétei komoly gondokat is előidézhetnek. A kialakult helyzetben mégsem tartjuk lehetségesnek és helyesnek az iskolák életébe való beavatkozást, ezért csak egy út marad a korszerű ismeretek befogadására: ha sikerül olyan társadalmi, gazdasági környezetet és ennek olyan pozitív ráhatását az iskolákban elérni, amely mintegy kikényszeríti és ösztönzi a korszerű, új ismeretek oktatását. Ehhez viszont olyan társadalmi nyomásra, közhangulatra van szükség, amely képes ezt a hatást elérni. Ehhez járul hozzá a központi célok és források koncentrált és hatékony megjelenése, amely egyrészt irányt ad, másrészt húzóerőt biztosít.

Mi úgy véljük, hogy ezek a körülmények fogják végső soron eldönteni a program jövőjét, és ezért semmiképpen sem tartjuk elfogadhatónak azokat az elképzeléseket, amelyek valamilyen központi utasítással, megyei tanácsoknak, művelődési szerveknek és iskoláknak szóló előírásokkal, szabályokkal, szankciókkal kívánják a program továbbhaladását biztosítani.

A program 5 éve alatt eddig is nagymértékben támaszkodtunk a pedagógusok, szakemberek véleményére, és a szükségesnek megfelelően módosítottuk elképzeléseinket, a program megvalósításával kapcsolatos terveinket. A pedagógusok részvételével eddig évente megrendezett 25—30 ilyen találkozó igen hasznos volt számunkra és erre az új követelményeknek megfelelően még nagyobb szükség lesz a jövőben.

Úgy gondoljuk, hogy az előbb elmondottak szellemében erőfeszítéseinket ezért elsősorban a közvélemény hatékony és folyamatos tájékoztatására, támogatásának megőrzésére, ezzel egyidejűleg további anyagi források biztosítására kell irányítanunk; a mainál lényegesen részletesebben, gyakrabban és jobban kell a közvéleményt, az iskolák környezetét és magát az iskolát tájékoztatni elképzeléseinkről, gondjainkról. Folytatnunk kell azt a gyakorlatot, amit már eddig is megvalósítottunk, hogy évről évre a legkülönfélébb országos, megyei és pedagógiai fórumokon tájékoztatást adjunk a programról, gyűjtsük össze a tapasztalatokat és nyissunk vitát az elképzelésekről.

PÁRIS GYÖRGY,
a TII igazgatója

TechnoMIR
Analóg modulok

Minden jel feldolgozható

A TechnoMIR interfészrendszer keretében kifejlesztett A/D modul olyan programozható analóg-digitál konverter, amely a bemenetére érkező analóg jelet (feszültséget) digitális értékké (adattá) alakítja.

A modul elvi vázlata és transzfer karakterisztikája az *1. ábrán* látható. A rendszeren belül két A/D modultípus létezik:

1. *TechnoMIR M. 2 A/D modul*

Ez a típus, melynek blokkvázlatát a *2. ábra* mutatja, a kompenzáció elve alapján működik. A vezérlő logika addig változtatja a D/A konverter bemenetét, amíg a feszültsége egyenlővé nem válik a bemeneti analóg jellel. Az egyenlőséget a komparátor jelzi.

Az átalakítás eredményét — méréshatártól függően — a következő utasításokkal olvashatjuk be:

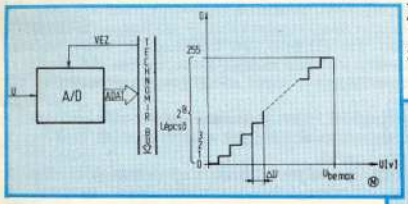
1 V INP(15)
10 V INP(14)
100 V INP(13)

A címek HT gépre érvényesek, más gépekre a megfelelő címkorrekcióra van szükség.

2. *TechnoMIR M. 2.1 A/D—2 modul*

A modul belső struktúráját a *3. ábrán* vizsgálhatjuk meg. Az előző megoldáshoz képest szembetűnő változás, hogy ennél az átalakítónál két darab D/A van beépítve: az egyik a mérő D/A konverter, a másik referenciaforrás.

További újdonság a differenciálerősítő és a programozható bemenet-kimeneti regisztersor. Ez utóbbit, akárcsak a PB modulban, 8255 típusú áramkörrel valósították meg.



1. ábra

Az A/D—2 modul átalakítási elve

A bemeneti differenciálerősítőre kapcsolt mérendő feszültséget a komparátor összehasonlítja a mérő D/A jelével, amely az A csatornán keresztül programozható. Egyezés esetén a komparátor a C regiszter legfelső bitjével jelez. Ekkor a mérendő jel digitális értéke a mérő D/A bemeneti értékével egyenlő, tehát az átalakítás eredménye az A regiszterben megtalálható.

Ebben az üzemmódban a modul −1,2 V-tól +1,2 V-ig képes feszültséget átalakítani, amit a +IN bemenetre kell kapcsolni, a +IN és a föld közé. Ugyanilyen feszültségtartományban, földfüggetlenül is működhet az átalakító, ha a +IN és a −IN bemenetek melegpontjai közé kapcsoljuk a mérendő feszültséget.

A modul másik üzemmódja a kibővített méréshatárú üzem. Ehhez fel kell használni a B regisztert keresztül programozható referencia D/A konvertert is. Ennek a módnak az a lényege, hogy a REF OUT kimenet és a −IN bemenet kívülről történő összekötésével az átalakítás vonatkoztatási pontját eltoljuk a földponttól a referencia D/A kimeneti feszültségének pontjára. Így az A/D modul nem a földhöz képest méri a feszültséget, hanem a programból beállítható szinttől.

Az áramkör adatait a *táblázat* tartalmazza. A címek HT gépre vonatkoznak.

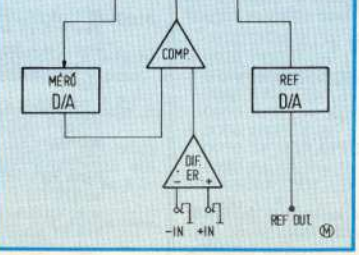
Kontrollregiszter	Utasítás: OUT 8,J
Címe: 11	(0<J<255)
Utasítás: OUT 11,136	Kimeneti feszültség
Mérő D/A átalakító	−10 V-tól +10 V-ig
Címe: 8	Maximális terhelő áram:
Utasítás: OUT 8,I	5 mA
(0<I<255)	**Komparátorfigyelés**
Referencia D/A átalakító	Címe: 10
Címe: 9	Utasítás: INP(10)

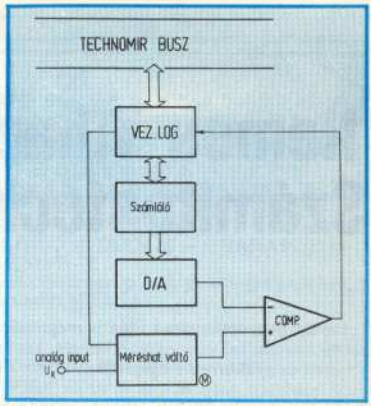
Feszültségmérés földhöz képest

A mérés a −1,2 V +1,2 V tartományban végezhető el. Mielőtt nekilátnánk, gondoljuk át, mit kell elvégeznie a mérőprogramnak:

1. A kontrollregiszter beállítása
2. Fűrészfeszültség előállítása a mérő D/A segítségével
3. A komparátor figyelése
4. Ha a C regiszter legfelső bitje (bit7) jelez, a mérő D/A érték kiolvasása
5. A mérési ciklus újraindítása

2. ábra





3. ábra

Maga a program az *1. listán* látható. Az 1. pont a 10-es sor, a 2. pont a ciklus, a 20, 30, 40-es sor. A 3. pont, a komparátorfigyelés, a 40-es sor IF utasításában történik.

Egy másik módszerrel lényegesen lerövidíthetjük a konverziós időt *(2. lista)*. Ez a fokozatosan közelítő eljárás (successive approximation). Lényege, hogy először a legfelső bitnek (bit7) megfelelő feszültséget állítjuk elő a D/A kimenetén. Ez éppen az intervallum fele. A komparátor segítségével eldöntjük, hogy a keresett feszültség értéke nagyobb-e ennél. Ha igen, ezt a bitet bekapcsolva hagyjuk, ha nem, kikapcsoljuk. Ezt ismételve, az alacsonyabb biteknél is a konverzió ideje maximum 8 ciklusidő.

Földfüggetlen feszültségmérés

A mérés elve, módszere és programja megegyezik az előzőekkel. A különbség a bekötésben van, itt nem a modul földpontjához képest mérünk, hanem a mérendő áramkörben lesz a vonatkoztatási pont.

```
10 OUT 11,136
20 FOR J=0 TO 255
30 OUT 8,J
40 IF INP(10) AND 128 <> 0 THEN NEXT J
50 PRINT@0,(J-128)*0.01;"V"
60 GOTO 20
```

1. lista

```
10 OUT 11,136
20 FOR K=7 TO 0 STEP -1
30 OUT 8,2 K+U
40 IF (INP(10) AND 128)<>0 THEN U=U+2 K
50 NEXT K
60 PRINT U/100-1.27
70 U=0
80 GOTO 20
```

2. lista

Kiterjesztett méréshatárú mérés

Az előző módszerekhez képest nemcsak a mérő D/A átalakítót kell vezérelni, hanem a frekvencia D/A-t is. Ennek a feszültségértékével toljuk el a vonatkoztatási pontot, megnövelve ezzel a mérhető maximális feszültség értékét. A mérési intervallum így is maximum 2,5 V, de ez lehet például 5,1 V-tól 7,6 V-ig.

Ezzel a jelenleg forgalomban lévő modulok bemutatását befejeztük. A következőkben a fejlesztés útjait mutatjuk be.

KIRÁLY LÁSZLÓ — ALBU LÁSZLÓ

Nemes Tihamér Országos Középiskolai Számítástechnikai Tanulmányi Verseny

Immár hagyományosan idén is megrendezték a középiskolások számítástechnikai versenyét. A verseny két fordulóból állt: írásbeli, elméleti és gyakorlati megméretésből. Az alábbiakban az első forduló feladatait, megoldásait és az értékelés szempontjait közöljük. Az idei versenyen 206 iskolából 1717 diák indult. Az első fordulóból 54-en jutottak tovább, ill. teljesítették a 34 pontos szintet. Az általános vélemény szerint az idei megméretés nehezebb volt az elmúlt évinél, amit nem is annyira a feladatok bonyolultsága, mint inkább a mennyisége okozott. Az általános tapasztalatokat, véleményeket egy későbbi számunkban adjuk közre.

1. Az $(x - 10.0 ** i) * (x - 1.0) = 0.0$ egyenlet gyökeit az

```
a := 1.0 ;
b := - (10.0 ** i + 1.0) ;
c := 10.0 ** i ;
x1 := (-b - SQRT (b ** 2 - 4.0 * a * c)) / (2.0 * a) ;
x2 := (-b + SQRT (b ** 2 - 4.0 * a * c)) / (2.0 * a)
```

utasításokkal akarjuk kiszámítani az $i = 1, 2, \ldots, 25$ értékekre, mégpedig egy olyan számítógépen, amelyen a valós számokat lebegőpontos alakban ábrázolják: a mantissza 24 bites bináris szám (az egyik bit az előjel), a 2-es alapú kitevő pedig 8 bites, ugyancsak bináris szám (az egyik bit itt is az előjel). A programban a ** a hatványozás jele.

Ha x1-et és x2-t kiiratjuk, milyen i értékek esetén nem kapjuk meg a helyes megoldást?

Elérhető pontszám: 6.

Megoldás, értékelés:

A b**2—4*a*c diszkriminánsban i növekedésével b**2 értéke sokkal gyorsabban nő 4*a*c értékénél, és a számábrázolás pontatlansága miatt a b**2 mellett a 4*a*c előbb-utóbb elhanyagolható értékűvé válik. 1 pont.

Ekkor x2 = —b+SQRT (b**2) = 0.0 lesz az eredmény a várt 1.0 helyett! 1 pont. A 24 bites mantisszával max. 2**23 = 8 388 608-at, azaz 10**6 nagyságrendű számokat lehet pontosan ábrázolni. 1 pont.

A kivonáshoz a kitevőknek egyenlőknek kell lenniük, ezért $i > 6$ esetén már fellép az említett jelenség. 1 pont.

Az adott számábrázolás mellett max. 2**(2**7) = 2**128 ~ 10**38 nagyságrendű számok ábrázolhatók. 1 pont.

Ezért túlcsordulás $b > 10**19$, vagyis $i > 19$ esetén lép fel. 1 pont.

2. Az alábbi program az m természetes számhoz megkeresi annak a leghosszabb [i, j] zárt intervallumnak a hosszát, amelyre az $i, i+1, \ldots, j$ természetes számok összege pontosan m ($1 <= i <= j$).

```
    hossz := 0 ;
[1] WHILE m > az első hossz szám összege
        REP hossz := hossz + 10 ENDREP ;
[2] WHILE m < az első hossz szám összege
        REP hossz := hossz - 1 ENDREP ;
[3] WHILE (m - az első hossz szám összege) MOD hossz <> 0
        REP hossz := hossz - 1 ENDREP.

    az első hossz szám összege:
        IF hossz MOD 2 = 0
        THEN hossz DIV 2 * (hossz + 1)
        ELSE (hossz + 1) DIV 2 * hossz
        ENDIF .
```

A programban hossz (=j—i+1) és m egész számokat jelölnek, DIV egészosztás hányadosát, MOD egészosztás maradékát adja eredményül, WHILE ... REP ... ENDREP elöl vizsgált ismétlés, az első hossz számösszege a kiszámított értéket átadó (a függvényeljárásokhoz hasonló) algoritmus.

2.1 Mi a szerepe az [1]-gyel, [2]-vel, ill. [3]-mal jelölt ismétléseknek?

2.2 Hányszor kell végrehajtani a [3] ismétlést 4095, ill. 4096 esetén?

2.3 Mutasd meg, hogy a program biztosan befejeződik (azaz nincs benne végtelen ciklus)!

2.4 Mutasd meg, hogy ha m 2 nem negatív hatványa, akkor a maximális intervallumhossz éppen 1!

2.5 Hogyan lehetne gyorsabbá tenni a programot a meglévő jó tulajdonságok megőrzésével?

Elérhető pontszám: 15.

Megoldás, értékelés:

2.1 [1] meghatározza azt a hosszt, amely mellett az [1,hossz—10] intervallumba eső számok összege még kisebb, az [1,hossz] intervallumba eső számok összege pedig már nagyobb m-nél. 1 pont.

[1] eredményéből kiindulva [?] azt a hosszt határozza meg, amely mellett az [1,hossz] intervallumba eső számok összege még éppen kisebb, az [1,hossz+1] intervallumba eső számok összege pedig már éppen nagyobb m-nél. 1 pont.

[2] eredményéből kiindulva [3] azt vizsgálja, hogy [az első hossz szám összege, m] intervallum felbontható-e hossz számú, egyenlő hosszúságú részintervallumra. Ha igen, megvan a megoldás, hiszen csak a részintervallumok hosszával kell megnövelni az [1,hossz] intervallum minden elemét. Egyébként hosszt 1-gyel csökkentjük, és folytatjuk. 2 pont.

2.2 4095 esetén egyszer sem, ugyanis 4095 éppen az első 90 természetes szám összege, és ezért [3] ismétlési feltétele már kezdetben hamis. 1 pont.

4096 esetén 89-szer, ugyanis 4096 2 hatványa, amelynél a keresett intervallum hossza 1 (l. a 2.4 választ). Mivel a [2] ismétlés végén hossz=90, a hossz=1 érték megtalálásáig éppen 89-szer kell megismételni a törzset. 1 pont.

2.3 Itt három ismétlés van, amelyekben — elvileg — elakadhatna a program. De [1] biztosan befejeződik, mert hosszat 10-esével növelve az első hossz szám összege előbb-utóbb biztosan nagyobb lesz a véges m-nél (vagy túlcsordulás lép fel). 1 pont.

De [2] is befejeződik, mert hosszat 1-esével csökkentve legfeljebb 9 lépésben m-nél megint kisebb lesz az első hossz szám összege. 1 pont.

Végül [3] is biztosan véget ér, hiszen a legrosszabb esetben is hossz=1-gyel minden szám maradék nélkül osztható. 1 pont.

2.4 Ha m felbontható az [i,j] intervallumba eső számok összegeként, akkor m = hossz*(i+j)/2 = hossz*(2*i+hossz—1)/2, hiszen j = i+hossz—1. A vizsgálandó esetben m=2**k alakú (ahol ** a hatványozás jele), vagyis tetszőleges k-ra teljesülnie kell a következő egyenlőségnek: 2**(k+1) = hossz*(2*i+hossz—1). De ha hossz páros, akkor 2*i+hossz—1 páratlan, és fordítva, tehát mindkét tényező egyszerre csak hossz=1 mellett lehet 2 hatványa. 3 pont.

2.5 A program gyorsabbá tehető, ha a szorzások és az osztások helyett összadást és kivonást végzünk. Erre van lehetőség, ugyanis az első hossz szám összegénél az első hossz+1 szám összege éppen (hossz+1)-gyel nagyobb. Ha bevezetünk egy változót az összeg tárolására, akkor ennek értékét az [1], [2] és [3] ismétlésekben minden lépésben meghatározott értékkel kell növelni, ill. csökkenteni. 2. pont.

Ha az adott programozási nyelven ábrázolható legnagyobb egész számot ismerjük, akkor tudunk felső korlátot adni a

hosszra, és így az [1] és [2] ismétlés helyett a gyorsabb bináris (logaritmikus) keresést használhatjuk. 1 pont.

3. Egy dobozban fekete és fehér kávészemek vannak. Véletlenszerűen kiveszünk két szemet. Ha ezek azonos színűek, akkor helyettük egy feketét rakunk vissza. (Erre a célra korlátlan mennyiségű fekete kávészem áll a rendelkezésünkre.) Ha viszont különböző színűeket húzunk, akkor egy fehéret dobunk vissza a dobozba.

3.1 Hogyan változik a fekete, ill. a fehér kávészemek száma a folyamat során?

3.2 A kávészemek számának milyen lényeges tulajdonsága nem változik meg az algoritmus végrehajtása közben?

3.3 Mitől függ, hogy milyen színű kávészem marad a dobozban?

Elérhető pontszám: 8.

Megoldás, értékelés:

3.1 A fekete szemek száma 1-esével nő, ha két fehéret húzunk, egyébként 1-esével csökken. 1 pont.
A fehér szemek száma 2-esével csökken, ha két fehéret húzunk, egyébként nem változik. 1 pont.

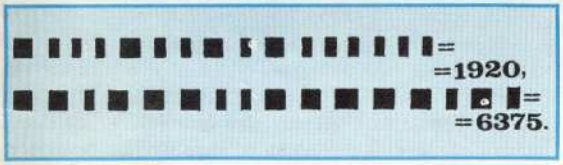
3.2 A fehér szemek számának párossága (ez ún. invariáns tulajdonság)! Ha kezdetben páratlan volt a fehér szemek száma, akkor végig páratlan is marad, ha páros volt, akkor páros is marad. 3 pont.

3.3 Minden lépésben 1-gyel csökken a kávészemek száma a dobozban. 1 pont.
A fehérek kezdeti számától függ, hogy milyen színű lesz az utolsó: ha páratlan, akkor fehér; ha páros, akkor fekete. 2 pont.

4. A Sakál üzletház pénztárgépeihez vonalkódolvasó készülékeket vásárolt. Néhány hónap alatt kiderült, hogy a készülékek nem a nemzetközileg elfogadott vonalkódot, hanem valami mást ismernek fel. Bit Benő elhatározta, hogy megfejti a kódot. A készülék gépkönyvében erről a következőket találta:

— A vonalkódban a bináris 0-kat egyszeres, az 1-eseket kétszeres vastagságú sötét vonal jelöli. A sötét vonalakat legfeljebb kétszeres vastagságú fehér csíkok választják el egymástól.

— Egy-egy árucikk azonosítója 1000..6999 közötti decimális szám; minden számjegyet 4 sötét vonallal kell kódolni. Példák:



— A kódolvasó ceruzát a papirra kell tenni, és a vonalakra lehetőleg merőleges irányban folyamatos kézmozgással kell végighúzni.

— Az árucikk száma nem függ attól, hogy a kódolvasó ceruzát elölről hátrafelé vagy fordítva húzzuk-e végig a kódolt számsoron.

4.1 Mit állapított meg Benő, mi a számjegyek kódja?

4.2 Hogyan ismerhető fel a cikkszám mindkét irányból?

Elérhető pontszám: 9.

Megoldás, értékelés:

4.1 A példákból elölről hátrafelé kiolvashatók a következő kódok: 0=0000, 1=1000, 2=0100, 3=1100, 5=1010, 6=1101, 7=1111, 9=1001. 2 pont.
Visszafelé olvasva néhány kód más: 1=0001, 2=0010, 3=0011, 5=0101, 6=1011. 1 pont.
Hiányzik a 4-es és a 8-as számjegy, amelyekhez a még hiányzó 0110, 0111 és 1110 kódszavak rendelhetők. 1 pont.
A hozzárendelés a 4.2 választól függ.

4.2 A számjegyek kódjából ki lehet deríteni az olvasási irányt, és így helyre lehet állítani az eredeti sorrendet. 1 pont.
Ehhez az szükséges, hogy a négy közül legalább egy számjegynek ne legyen szimmetrikus a kódja: ezért írják elő, hogy a legnagyobb helyi értékű jegy csak 1,2,3,4,5 és 6 lehet. 2 pont.
Ebből következik, hogy a 4 számjegynek két kódja van: 0111 és 1110, a 8-nak pedig egy: 0110. 2 pont.

5. Mit csinál a következő program, és milyen bemenő adatokra működik rosszul (feltéve, hogy a$-ba számjegyeket, b-be pedig pozitív egész számot írunk)?

```
10 INPUT "Szamjegyeket kerek";a$
20 INPUT "Pozitiv egesz szamot kerek";b
30 IF LEN(a$) < 3 OR b >= LEN(a$) THEN 10
40 PRINT MID$(a$,1,1);".";MID$(a$,2,b-2);
50 c$ = MID$(a$,b,1)
60 IF MID$(a$,b+1,1) > "5" THEN c$ = CHR$(ASC(c$)+1)
70 PRINT c$;"E";LEN(a$)-1
```

MID$(x$,p,h) az x$ karaktersorozat p-edik pozícióján kezdődő, h hosszúságú karaktersorozatot állítja elő.

Elérhető pontszám: 6.

Megoldás, értékelés:

A beolvasott decimális számot normalizálja. 1 pont.
A számot b db számjeggyel írja ki. 1 pont.
A többi számjegyet kerekíti. 1 pont.
Ha a$ 0-val kezdődik, akkor hibásan normalizál. 1 pont.
Ha az utolsónak kiírandó számjegy 9-es lenne, és ezt a kerekítés miatt meg kell növelni, akkor nem számjegyet ír ki, hanem a 9-es után következő kódú karaktert. 2 pont.

6. Adott egy 0-kból és 1-esekből álló számsorozat, meghatározandó a csupa 1-esekből álló szakaszok hossza. Pl. a 0 0 0 1 1 1 0 0 1 1 0 0 sorozatban van egy 3 és egy 2 hosszúságú 1-es szakasz. Két algoritmust készítettünk a megoldásra, mindkettő az A(N) vektort használja. (Feltesszük, hogy N>1 természetes szám, és az A(N) vektor csak 1 és 0 elemeket tartalmaz.)

```
A: I := 2 : H := 0
   Ciklus amíg I <= N
      Ha A(I-1)=0 és A(I)=1 akkor
                  H := 0
                  Ciklus amíg a A(I)=1
                     I := I+1 : H := H+1
                  Ciklus vége
                  Ki: H
      különben
                  I := I+1
      Elágazás vége
   Ciklus vége

B: I := 1 : H := A(1)
   Ciklus amíg I<N
      Ha A(I)=0 és A(I+1)=1 akkor H := 0
      Ha A(I+1)=1 akkor H := H+1
      Ha A(I)=1 és A(I+1)=0 akkor Ki: H
      I := I+1
   Ciklus vége
```

6.1 Milyen esetben működnek különbözően és mi ez a különbség?

6.2 Milyen esetben hibás valamelyik és mi a hiba?

Elérhető pontszám: 7.

Megoldás, értékelés:

6.1 A és B másképpen működik, ha a számsorozat 1-esekkel kezdődik vagy végződik. 1 pont.
Ekkor csak B veszi figyelembe az első sorozat hosszát. 2 pont.

6.2 Mindkettő hibás, ha a számsorozat 1-esekkel végződik. 1 pont.
Ekkor B kiszámolja, de nem írja ki a megoldást. 2. pont.
Ekkor A-ban a belső ciklus feltételében indexhatár-túllépés lesz. 1 pont.

7. Mit rajzol a következő COMAL nyelvű program, ha feltesszük, hogy a rajzolás során végig a képernyőn maradunk?

```
0010 DIM X(5), Y(5)
0020 FOR I := 1 TO 4 DO
0030 INPUT X(I), Y(I)
0040 ENDFOR I
0050 X(5) := X(1); Y(5) := Y(1)
0060 INPUT A
0070 SETGRAPHIC 0
0080 ALFA := 0
0090 RAJZOLAS
0100 ALFA := A
0110 RAJZOLAS
0120 STOP
```

```
0200 PROC RAJZOLAS
0210 PENUP
0220 SETXY X(1) * COS(ALFA) + Y(1) * SIN(ALFA),
          -X(1) * SIN(ALFA) + Y(1) * COS(ALFA)

0230 PENDOWN
0240 FOR I:= 2 TO 5 DO
0250 SETXY X(I) *COS(ALFA) + Y(I) *SIN(ALFA),
          -X(I) *SIN(ALFA) + Y(I) *COS(ALFA)

0260 ENDFOR I
0270 ENDPROC RAJZOLAS
```

SETXY x,y jelentése: húzz egyenest az aktuális ponttól (x,y)-ig, majd vedd (x,y)-t az aktuális pontnak. SETGRAPHIC 0 grafikus üzemmódba vált át, PENUP felemeli, PENDOWN pedig leteszi a tollat.

Elérhető pontszám: 7.

Megoldás, értékelés:

A 90-es sorbeli RAJZOLAS a koordináták által kijelölt négyszöget rajzol a képernyőre. 2 pont.

A 110-es sorbeli RAJZOLAS hasonló, de ALFA szöggel elforgatott négyszöget rajzol a képernyőre. 4 pont.

Az utóbbi a (0,0) pont mint középpont körül van elforgatva. 1 pont.

8. Egy n*m-es tömbben tároljuk az ország térképét. Minden (x,y) koordinátájú pontban M(x,y) jelenti az adott pont tengerszint feletti magasságát. A következő BASIC programrészlet megpróbálja megkeresni az ország legmagasabb pontjának koordinátáit, mindig a legmeredekebb növekedés irányában haladva.

Milyen esetekben nem találja meg, és mi történik ilyenkor?

```
100 x = RND(n) : y = RND(m) : REM 1 és n, ill. m közötti
    véletlen egész
110 x0 = x : y0 = y
120 FOR i = x-1 TO x+1 : IF i < 1 OR i > n THEN 160
130 FOR j = y-1 TO y+1 : IF j < 1 OR j > m OR (i = x
    AND j = y) THEN 150
140 IF M(i,j) >= M(x0,y0) THEN x0 = i : y0 = j
150 NEXT j
160 NEXT i
170 IF x0 <> x OR y0 <> y THEN x = x0 : y = y0 :
    GOTO 110
180 PRINT x,y
```

Elérhető pontszám: 9.

Megoldás, értékelés:

A program befejeződik, ha olyan pontot talál, amely minden szomszédjánál magasabb, de mégsem a legmagasabb. 2 pont.

Ha fennsíkra ér, végtelen ciklusba kerül. (A fennsík egynél több azonos magasságú pontból áll, amelyeket alacsonyabb pontok vesznek körül.) 3 pont.

Síkon haladva mindig „délkeleti" irányba törekszik. 1 pont.

Akkor is végtelen ciklusba kerül, ha egy síkon haladva alacsonyabb pontokba ütközik (2 pont), vagy eléri a térkép szélét (1 pont).

9. Egy lánctalpas játékautóra 2 fényérzékelőt szereltünk, amelyek egymással 10-15 fokos szöget zárnak be. A jármű bal, ill. jobb oldali lánctalpát külön is, egyszerre is működtethetjük. A járműnek adható parancsok a következők:

OUT 10, szám a szám legkisebb helyi értékű bitje a bal oldali, a következő pedig a jobb oldali lánctalp mozgatására használható (1 = elindul, 0 = megáll);

INP(11) értéke a bal oldali fényérzékelő által mért fényerősség (0—255 közötti egész);

INP(12) értéke a jobb oldali fényérzékelő által mért fényerősség (0—255 közötti egész).

A fényerősség mért értéke attól függ, hogy milyen messze van az érzékelő a fényforrástól és milyen szöget zár be vele. Készítettünk a játékautóhoz egy BASIC programot:

```
10 INPUT e : IF e <= 0 THEN 10
20 x = INP(11) - INP(12)
30 IF x > e THEN 100
40 IF x < -e THEN 200
50 OUT 10,3
60 IF ABS(INP(11) - INP(12)) <= e THEN 60
70 OUT 10,0
80 GOTO 20
100 OUT 10,2
110 IF INP(11) - INP(12) > e THEN 110
120 OUT 10,0
130 GOTO 20
200 OUT 10,1
210 IF INP(12) - INP(11) > e THEN 210
220 OUT 10,0
230 GOTO 20
```

9.1. Hogyan mozog a játékautó a padlón, ha az álló fényforrást a fényérzékelőkkel egy magasságban helyezzük el?

9.2. Mi az e változó szerepe?

9.3. Álló, ill. mozgó fényforrás mellett milyen esetben lehetnek gondok a program működésével?

Elérhető pontszám: 8.

Megoldás, értékelés:

9.1. Törtvonalban vagy egyenes vonalban halad a fényforrás felé. 2 pont.

De ha a fényforrás az autó mögött van, távolodik tőle. 1 pont.

9.2. Az e a fényerősség-különbségre vonatkozó érzékenységi küszöb. 1 pont.

9.3. Ha e olyan kicsi, hogy az autó a lehető legkisebb elfordulás után is azonnal az ellenkező irányba akar fordulni (azaz nem tud elég kicsit fordulni). 2 pont.

Ha e olyan nagy, hogy az autó állandóan elkerüli a fényforrást. 1 pont.

Ha a fényforrás olyan gyorsan mozog, hogy a játékautó nem tud utánafordulni. 1 pont.

Az NJSZT Nemes Tihamér Országos Középiskolai Számítástechnikai Tanulmányi Verseny versenybizottsága kérésére közöljük az alábbi nyílt levelet:

Tisztelt Tanár Kollégák!

Az 1988. évi Nemes Tihamér Országos Számítástechnikai Tanulmányi Verseny első fordulója — szándékunk ellenére — túlságosan nehézre sikerült. A versenyen résztvevő 1717 tanulónak mindössze 80 dolgozatát küldték be a versenybizottságnak. Tudjuk, hogy a sikerélmény ilyen mértékű hiánya károkat okozhat a versenynek, visszavetheti a tanulók érdeklődését. Ezt a jövőben mindenképpen el szeretnénk kerülni. Ezért szükségesnek tartjuk felhívni a figyelmüket arra, hogy a következő évek versenyein a feladatok nehézségének, ill. mennyiségének meghatározásában az 1987. évből indulunk ki. Célunk, hogy minden évben kb. 400-500 dolgozat kerüljön a bizottság elé a számítástechnikai kultúra fejlődésének megfelelően emelkedő szinten. Ezúton is szeretnénk megköszönni iskolájuk részvételét a versenyben, valamint az Önök lelkiismeretes munkáját a dolgozatok javításában.

A versenybizottság

Zenei képességek vizsgálata számítógéppel

C64

Hallásérzékenység mérése

A program két egymáshoz közel álló hangot generál (120-150-es sorok) ciklusban. Arra kell válaszolni, hogy a második hang mélyebb-e vagy magasabb az elsőnél. Bár a választ már az első hang után lehet tudni (120-as sor), mégse kapkodjunk (170-es sor), és győződjünk meg arról, hogy a szándékunk szerinti billentyűt nyomtuk-e meg. A sorozat 32 ezrelékes frekvenciaváltozástól 1,2 ezrelékig tart (70, 190, 270-es sorok). Teljesítményünket a véletlen szerencse vagy nyilvánvaló tévedés is befolyásolhatja, ezért eleinte érdemes háromszor végigpróbálni a programot (40, 310-es sorok), hogy reális képet kapjunk (380-as sor).

Néhány szempont a kiértékeléshez: egy temperált félhang (például zongorán e, f) kb. 60 ezrelékes hangmagasság-változást jelent. Az átlagos hallás azonban ennél kisebbet is érzékel. Elfogadható, ha az átlag 12 ezrelék körül van. Kissé kihúzhatjuk magunkat, ha az első átlagunk 6—8 ezrelék lesz.

Évekig lappangó zenei tehetségünk számszerű bizonyítékát vélhetjük felfedezni a 3 ezrelék körüli átlagban. Azonban jóképességű zongoristák 2, vonósok 1,2 ezreléket is el fognak érni.

Ha valaki a 70-es sorban a ciklus felső határát maximálisan 2-re növeli, s ebben az esetben eljut 0,6-0,8 ezrelékes átlagig, az nyugodtan foglalkozhat a szoborkészítés gondolatával, vagy áttérhet a következő számban megjelenő programra.

KALMÁR GYULA

```
10 PRINT"■":S=54272:FORI=0TO24
20 POKES+I,0:NEXT
30 POKES+24,15:POKES+6,250
40 P=P+1:Q=INT(RND(0)*15)+15:Q=360*Q
50 PRINT"MAGASABB(+),VAGY"
60 PRINT" MELYEBB(-) LETT A HANG?"
70 FORI=4TO-1.2 STEP-.4:K=0
80 IFI=4THENQ(0)=Q
90 Y=INT(RND(0)*10)+1
100 IFY>4THENY=1:GOTO120
110 Y=-1
120 K=K+1:C=INT(Q(K-1)*(1+Y*2↑I/1000))
130 Q(K)=C:F=INT(C/256):A=C-256*F
140 POKES,A:POKES+1,F:POKES+4,17
150 IFK=2THENQ(0)=Q(2):POKE198,0:GOTO170
160 FORT=1TO1000:NEXT:GOTO120
170 FORT=1TO300:NEXT:POKES+4,16
180 GETT$:IFT$=""THEN180
190 HI=INT(10*2↑I+.5)/5
200 IFT$="+" ANDQ(1)/Q(2)<1THEN250
210 IFT$="-" ANDQ(2)/Q(1)<1THEN250
220 PRINT"■ TEVEDTEL.A HANG  VALTOZASA"
230 PRINTHI"EZRELEK VOLT"
240 GOTO310
250 PRINTTAB(19)"-JO-":FORT=1TO400:NEXT
260 PRINT"■"TAB(19)"     ":PRINT"■"
270 NEXT
280 PRINT"■HALLASOD RENDKIVUL JO"
290 PRINTTAB(9)HI"EZRELEK"
300 PRINT"VALTOZAST VETTEL ESZRE"
310 AT=AT+HI:IFP=3THEN370
320 PRINT"■FOLYTASSUK?(I/N)"
330 GETT$:IFT$=""THEN330
340 IFT$="I"THENPRINT"■":GOTO40
350 IFP>1THEN370
360 GOTO390
370 AT=AT/P:AT=INT(10*AT+.5)/10
380 PRINT"■AZ ATLAG:"AT"EZREL"
390 END
```

Új sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihamér országos számítástechnikai verseny színvonalának felelnek meg. Minden esetben olyat választunk, amely röviden, gyorsan megoldható, de megoldásához ötletre van szükség. A megoldást mindig a következő számban közöljük.

Mivel változatosságra törekszünk, különböző programozási nyelveket használunk. Az is előfordul majd, hogy egy feladatra több programnyelven is közlünk megoldást, ezzel is elősegítve az ismeretszerzést.

A szerkesztőség várja az olvasók, a versenyzők leveleit. A legötletesebb program beküldőjét könyvutalvánnyal jutalmazzuk. Ne feledjenek azonban a program mellé leírást is mellékelni!

1. feladat. Önkiíró

Írjon olyan programot, amely kiírja önmagát, de saját forrásszövegéhez semmilyen külső adatmezőhöz vagy perifériához nem fér hozzá és listájában nem használ láthatatlanná tevő trükköket! Ez BASIC-ben például azt jelenti, hogy PEEK, POKE, INPUT, LOAD, MERGE, LIST utasításokat, valamint speciális kontrollkaraktereket nem tartalmazhat és RUN-nal indul. READ, DATA utasítás használható.

Törekedjenek arra, hogy a program a nyelv és a számítógép minél kevesebb specialitását használja ki. Sok sikert a megoldáshoz!

PINTÉR GÁBOR

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 810-950/473

Elrejtett változók

Ennek a programnak a segítségével nagyon sok memóriaterületet spórolhatunk meg, tehát nem megy a programunk rovására. A C64-en folytatott programozás során, különösen, ha szerkesztett képernyőkkel, párbeszédes üzemmódban dolgozunk, igen sok üzenetet, tájékoztatást kell a kezelővel közölni.

Az úgynevezett üzeneteket tárolhatjuk magában a programban is, vagy külön fájlban, amelyet a programmal a memóriába, egy tömbbe beolvasva használunk. Mind-

1. lista

```
100 REM **** 1. LISTA ****
110 REM **** KIÍROL ****
120 SYS9*4096
130 .OPT OO
140 **=828
150 CIM = $FB
160 DB = $14
170 SZAML = $FD
180 VESSZO = 44
190 STRCIM = $62
200 STRH = $61
210 ;
220 SEI ;          BASIC VEKTOR ATIRASA
230 LDA #<KIÍROL
240 LDX #>KIÍROL
250 STA $30A
260 STX $308
270 CLI
280 RTS
290 ;
300 KIÍROL LDA #0 ;    A KIJELOLT 'A' JEL
310 STA $0D ;          VIZSGÁLATA
320 JSR $0073
330 CMP #"A"
340 BEQ KIÍR ;         HA EGYENLO KIÍRAS
350 JSR $0079 ;        HA NEM, VISSZA AZ
360 JMP $AE8D ;        INTERPRETERHEZ
370 ;
380 KIÍR JSR $0073 ;     A KOV. KARAKTER
390 JSR $AEFA ;     NYITO ZAROJEL VIZSG.
400 JSR $AD9E ;     ERTEK ATVETELE
410 JSR $B7F7 ;     ATALAKITSA 2 BAJTTA
420 LDA $1 ;     PROCESSZOR PORT ALLAPOTA
430 PHA ;           A VEREMBE
440 LDY #0 ;     A BASIC ALATTI TERULET
450 LDX #$A0 ;     KEZDOCIMENEK
460 STY CIM ;     BETOLTESE CIM-BE
470 STX CIM+1
480 STY SZAML ;     A SZAMLALO REGISZTER
490 STY SZAML+1 ;     NULLAZASA
500 LDA #$36 ;     ATKAPCSOLAS RAM-RA
510 STA $1
520 KER LDY #0 ;     AZ ELSO VESSZO JEL
530 KERI LDA (CIM),Y ;     KERESESE
540 CMP #VESSZO
550 BEQ NOVEL ;     HA MEGVAN EZ A KEZDET
560 INY ;              ---KERESES EDDIGIEN
570 BNE KERI ;              255 KARAKTERIG
580 CLC ;           HA NINCS VESSZO
590 BCC VISSZA2 ;     NINCS STRING, VISSZA
600 ;               AZ INTERPRETERHEZ
610 NOVEL INY ;        A MEGTALALT VESSZO
620 TYA ;           ATLEPESE ES HOZZAADASA A
630 CLC ;           KEZDOCIMHEZ, MERT EZ LESZ
640 ;
650 ;
```

```
645 ;
646 ; **** 1. LISTA FOLYTATAS ****
647 ;
650 ADC CIM ;     AZ ELSO TENYLEGES
660 STA CIM ;          STRING KEZDETE
670 BCC KN
680 INC CIM+1
710 KN INC SZAML ;              A NULLAZOTT
720 BNE HASON ;          2 BAJTOS SZAMLALO
730 INC SZAML+1 ;       NOVELESE EGGYEL
740 ;
750 HASON LDA SZAML ;        A MEGTALALT
760 CMP DB ;       STRING DARABSZAMA ES A
770 BNE KER ;        KIVANT STRING SORSZAM
780 LDA SZAML+1 ;          OSSZEHASONLITASA
790 CMP DB+1 ;       HA NEM EGYEZIK TOVABBI
800 BNE KER ;        KERES, HA IGEN, AKKOR
810 LDY #0 ;          A STRING VEGENEK
820 MEGKER LDA (CIM),Y ;          KERESESE
830 CMP #VESSZO ;     HA 255 KARAKTER UTAN
840 BED VISSZA ;        SEM TALALJA A VEGET
850 INY ;          JELZO VESSZOT, MAJDNEM NINCS
860 BNE MEGKER ;          HOSSZABB STRING
870 CLC ;          HOSSZU STRINGET
880 BCC VISSZA2 ;          AD VISSZA
900 VISSZA PLA ;          AZ ALLAPOT VISSZA
910 PHA ;          A VEREMBE
920 STA 1 ;          AZ ALLAPOT BEALLITASA
930 TYA ;          A MEGTALALT HOSSZUSAGU
940 JSR $B47D ;          STRING LETREHOZASA
950 LDY #0
960 LDA #$36 ;          VISSZA A RAM
970 STA 1 ;          ALLAPOTHOZ
980 ;
990 VIDI LDA (CIM),Y ;     AZ UJ STRING
1000 STA (STRCIM),Y ;          FELTOLTESE
1010 INY ;          A BASIC ROM ALOL
1020 CPY STEH
1030 BNE VIDI
1040 ;
1050 PLA ;          ROM VISSZAALLITASA
1060 STA 1
1070 CLC
1080 BCC VEGE ;     MAJD VISSZA BASICBE
1090 ;
1100 VISSZA2 PLA ;          A ROM ALLAPOT
1110 STA 1 ;          VISSZAALLITASA
1120 LDA #0 ;       NULLA HOSSZU STRING
1130 JSR $B47D ;          LETREHOZASA
1140 ;          A VEGLEGES STRING HELYRE-
1150 VEGE JSR $B8C6 ;          MASOLASA
1160 JMP $AE7F ;          VISSZA AZ
1170 ;               INTERPRETERHEZ
```

```
1 REM *** 2. LISTA ***
2 REM **** KIÍROL BASIC BETOLTO ****
3 REM HASZNALAT:  X$=&(SORSZAM)
4 S=0:C=828:I=0
5 READA:IFA=-1THEN7
6 POKEC+I,A:I=I+1:S=S+A:GOTO5
7 IFS<>20503THENPRINT"■■■■■HIBAHTHIBA"
8 END
9 SYS828:PRINT"■■■■■SIKERES■BEOLVASAS":END
10 DATA 120,169,73,162,3,141,10,3
11 DATA 142,11,3,88,96,169,0,133
12 DATA 13,32,115,0,201,38,240,6
13 DATA 32,121,0,76,141,174,32,115
14 DATA 0,32,250,174,32,158,173,32
15 DATA 247,183,165,1,72,160,0,162
16 DATA160,132,251,134,252,132,253,132
17 DATA 254,169,54,133,1,160,0,177
18 DATA 251,201,44,240,6,200,208,247
19 DATA 24,144,72,200,152,24,101,251
20 DATA 133,251,144,2,230,252,230,253
21 DATA 208,2,230,254,165,253,197,20
22 DATA 208,219,165,254,197,21,208,213
23 DATA 160,0,177,251,201,44,240,6
24 DATA 200,208,247,24,144,29,104,72
25 DATA 133,1,152,32,125,180,160,0
26 DATA 169,54,133,1,177,251,145,90
27 DATA 200,196,97,208,247,104,133,1
28 DATA 24,144,8,104,133,1,169,0
29 DATA 32,125,180,32,202,180,76,247
30 DATA 174,-1
```

2. lista

két esetben az üzenetek az amúgy sem túl bőséges BASIC területből vesznek el helyet, megröviditve ezzel a programot vagy a hasznos adatterületet. A gépben ugyanakkor van egy elég nagy, a BASIC-ből ki nem használható terület, amelyre ezek az adatok elhelyezhetők: a $A000-tól $D000-ig terjedő rész, amelyet csak BASIC kiegészítéssel lehet elérni. Ez az általam készített kis kiegészítés lehetővé teszi, hogy az erre a 12 k-nyi területre elhelyezett sztringeket BASIC-ből mint egyszerű sztringváltozókat használjuk.

Az assembler programot az *1. lista* tartalmazza, a működésére vonatkozó megjegyzésekkel együtt. A *2. lista* a BASIC-betöltő.

A program a kazettapufferben helyezkedik el, de tetszőleges betöltési címre átírható a BASIC-betöltő CÍM értékének megváltoztatásával. A kiegészítés SYS CÍMmel indítható.

A $A000—$D000 területre írt adatok kiolvasása BASIC-ből: X$=& (n), ahol X$ tetszőleges, a továbbiakban felhasználandó sztring, n pedig a kiolvasni kívánt adat sor-

TOP-lista

Listánkat felhasználói illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX—Spectrumra, Enterprise-ra, TVC-re, ATARI-ra és IBM-re készült programrangsorokat várunk havonta.

Címünk: 1371 Bp., Pf. 433. Mikroszámítógép Magazin Szerkesztősége.

Diákszerkesztőség

Felhasználói programok (C64-re)

1. ROCKMONITOR
2. SPEED PACKER
3. BURTS NIBBLER
4. ART STUDIO V 2.0
5. S-MON
6. MICRO RHYTM
7. CHEAT VIC
8. FUTURE WRITE II
9. DISC MANAGER
10. FILE HACK'EM

Játékprogramok (C64-re)

1. SPACE GAMES
2. SKATE OR DIE
3. SUPER HANG-ON
4. BUGGY BOY
5. NEBULUS
6. RASTAN SAGA
7. DRILLER
8. MANIAC MANSION
9. GOLDRUNNER
10. WINTER OLYMPICS '88

száma. A & (n) kifejezéssel hívott sztringgel az összes sztringművelet szabályosan elvégezhető. Például PRINT & (12) a képernyőre írja a 12. adatot.

A bevitt adatokban tetszőleges karakterek, így kurzorvezérlő jelek is lehetnek, kivéve a "," (vessző), mivel ez az egyes elemek elválasztására szolgál. Egy sztring itt is maximum 255 karaktert tartalmazhat.

Mint ebből is látható, az egyes üzenetek elválasztására a vessző (kódja 44) szolgál.

Kiváló tulajdonsága az ezen a területen elhelyezett adatnak, hogy a BASIC terület

```
100 REM *** 3. LISTA ***
110 REM **** ADATBEVITEL A BASIC ALA
120 REM ES KIMENTES ****
130 GOSUB470
140 INPUT"[RVS]A DARABSZAM: ";DB
150 A=40960:POKE,44
160 FORI=1TODB:PRINTI;:INPUT",",ADAT:A=
170 H=LEN(A$)
180 FORJ=1TOH
190 POKE<A+J>,ASC(MID$(A$,J,1))
200 NEXT
210 A=A+H+1:POKEA,44:NEXT
220 REM *** KIMENTESE ****
230 INPUT"[RVS]AZ EGYSEG SZAMA ";E
240 INPUT"[RVS]AZ ADATOK MEGNEVEZESE: ";M$
250 A=A+1:B1=INT(A/256):B2=A-(256*B1)
260 B2=A-(256*B1):PRINTA,B1,B2
270 PRINT"IS 53100"CHR$(34)"8"+
280 PRINT#CHR$(34)","E",1,40960,"A
290 POKE198,4:POKE631,19:POKE632,13
300 POKE633,13:POKE634,13:END
310 REM *** SAVE-ALL RUTIN ***
320 REM ** MINDEN TERULETROL
330 REM ** KEPES KIMENTENI
340 REM ** MIND SZALAGRA, MIND
350 REM ** LEMEZRE.
360 REM ** HASZNALATA:
370 REM ** SYS(C)"FILENEV",E,1,K,V+1
380 REM ** AHOL C = 53100 E=ESZKOZSZAM
390 REM ** K=KEZDOCIM V=VEGCIM
400 DATA 32,212,225,32,147,207,165,20
410 DATA 72,165,21,72,32,147,207,166
420 DATA 20,164,21,104,133,21,104,133
430 DATA 20,169,54,133,1,169,20,32
440 DATA 95,225,169,55,133,1,96,32
450 DATA 253,174,32,138,173,32,247,183
460 DATA 96,-1
470 C=53100
480 READA:IFA=-1THENRETURN
490 POKEC+I,A:I=I+1:GOTO480
```

3. lista

törlésekor, új BASIC vagy más helyen lévő program betöltésekor sem törlődik. Még a hardver reset sem képes törölni. Az ide egyszer beírt adat a gép kikapcsolásáig vagy szándékos felülírásig megmarad.

Az említett területre a *3. listán* látható programmal lehet adatokat bevinni és a bevitt adatokat adathordozóra kivinni. Ez a program tartalmazza az adatok adathordozóra való kimentésére szolgáló részt is, amely közönséges programfájlba menti ki az adatot. Ennek visszatöltésekor ügyelni kell, hogy ugyanerre a területre kerüljön vissza, vagyis a LOAD"prgnév",1,1 vagy LOAD"prgnév",8,1 utasítást kell használni.

Adatkimentésnél a kazettás egység száma természetesen a szokásos 1, a lemezegység száma 8. Megnevezésnek maximum 16 karakter használható.

Az adatbeviteli programban használt gépi kódú rész (200-tól 290-ig) más programokban tetszőleges területről kimentőként használható. Önálló alkalmazásakor a szintaxis a következő:

SYS 53100"fájlnév",eszközszám,1,kezdőcím,végcím + 1

KOCSIS GÁBOR

ROM-lista Primóra

Minden mikrogépnek van ROM-ja. Az ebben elhelyezkedő rutinokat a gép operációs rendszere (OS) használja. Az alábbi rutinokat a Primo assemblerében járatosaknak ajánljuk: saját programjukból meghíva használhatják eredményesen.

Decimális cím	A rutin leírása
457	CLS utasítás
467	RND rutin. Az A regiszterben előállít egy 0-255 számot
565	Magnóról egy bájtot beolvas az A regiszterbe
810	Az A regiszter tartalmának kiírása
865	INLINE szubrutin. Billentyűzetről max. 240 karakter beolvasása a HL által mutatott puffercímre, amelynek méretét a B regiszter mutatja
1740	BASIC megindítása
1746	Az RST vektorok és a periféria-ellenőrző táblázat a 16384-16426 címre kerülnek
2804	Adattípus-vizsgálat, ha nem szöveg-változó ➡ TM error
2806	TM-hiba
3015	HL—DE ➡ HL } túlcsordulás esetén egyszeres pontosságúra vált
3026	HL + DE ➡ HL }
3058	HL•DE ➡ HL }
3163	Egész szám túlcsordulás-vizsgálata, a túlcsordulás egyszeres pontosságúra vált
4007	BREAK esetén IN és a sorszám kiírása (ez ERROR esetén is érvényes)
4015	HL tartalmának kiírása számokkal
5640	BASIC függvények ugrótáblája
5712	BASIC kulcsszavak listája (a szavak első karakterének 7. bitje 1-re van állítva)
6178	BASIC utasítások ugrótáblája
6429	Szövegek sorvégjellel „ERROR"
6436	„IN"
6448	„BREAK"
6522	OM-hiba
6774	Programbevitel, utasítások kódolása, új sor beillesztése, mutatók beállítása
6956	A DE-ben levő sorszámú sor keresése, majd a sor címe DE-be kerül
7091	„?" kiírása majd INLINE
7104	Programszöveg-elemzés
7579	A billentyűzet vizsgálata LIST, programvégrehajtás közben
7593	STOP rutin
7652	CONT rutin
7671	TRON belépési pont
7672	TROFF belépési pont
7754	FC hiba
7857	GOSUB rutin
7874	GOTO rutin
7897	UL hiba
8568	„? REDO" szöveg
8602	INPUT rutin
8886	NEXT rutin
10406	Üzenetkiíró rutin, HL puffercím. 0-nál megáll
11009	Kifejezés egész értékének kiszámítása = DE
11049	— LIST belépés
11054	LIST rutin — LIST kilépés

SZARKA ZOLTÁN

Segédvonal

Ez a rutin a C64-en a programbevitelnél segít: áttekinthetőbbé teszi a sorokat, kisebb az esély a hibázásra.

```
0 REM ******** MATYASI  ARNOLD ********
1 REM ****** INDITASA: SYS 49152 ******
2 REM ***** JANOSHALMA,1988.11.1. *****
10 FORI=49152TO49265:READA:N=N+A
11 POKE I,A:NEXT I
12 IFN<>13328THENSTOP
13 SYS 49152
20 DATA 120,173, 14,220, 41,254,141, 14
21 DATA 220,173, 26,208,  9,  1,141, 26
22 DATA 208,169, 49,160,192,141, 20,  3
23 DATA 140, 21,  3,173, 17,208, 41,127
24 DATA 141, 17,208,234, 32, 97,192,234
25 DATA 133,255,169,255,141, 25,208, 88
26 DATA  96,169,242,141, 18,208,173, 25
27 DATA 208, 41,  1,141, 25,208,169,  2
28 DATA 160, 92,162,  0,141, 32,208,141
29 DATA  32,208,136,208,250,142, 32,208
30 DATA 142, 33,208,165,214, 42, 42, 42
31 DATA  24,105, 50,141, 50,192, 76, 49
32 DATA 234,165,214, 42, 42, 42, 24,105
33 DATA  50,141, 50,192,169,  7,141,134
34 DATA   2, 96
```

Sok bosszúságot okoz, ha a programbeírás közben elnézünk valamit, és csak később vesszük észre. Ilyenkor a programban keressük a hibát, de a képernyőt nézve hamarosan összefolynak a betűk, a hibát pedig nem találjuk. Ebben a munkában segít ez a program úgy, hogy a kurzor sorában egy vörös csíkot húz a képernyő széléig. Ez a csík a kurzorral mozog fel és le, tehát már a programbevitel közben szembe tűnik az a sor, amelyet éppen írunk. Ezáltal csökken a hibázás lehetősége. Ha elírunk valamit, könnyen javíthatjuk.

Gépeljük be figyelmesen a programot. Ha kész, akiknek nincs monitoruk BASIC-ből a SAVE utasítással, mások pedig mentsék ki a $C000—C072 tárcímet. Futtatás után, ha pontosak voltunk, a kívánt eredmény megjelenik, ha nem, nézzük át ismét nagyon figyelmesen, javítsuk ki a hibákat, végül indítsuk újra. Azoknak, akik sok programot írnak be papírról a gépükbe, megéri a fáradságot!

MÁTYÁSI ARNOLD

Könyvtárprogram C64-re

Bizonyára senki sem tudna kielégítő választ adni arra a kérdésre, hogy hány könyve van otthon, ki írta őket és mi a címük. Használjuk fel a kérdés megválaszolására a számítógépet! Segítségével bármely könyvünkről pontos információt kaphatunk.

A programmal *(1. lista)* rendszerezhetjük könyvtárunkat, hiszen minden egyes könyvről a címén és íróján kívül a kiadó nevét és a kiadási évet is tárolhatjuk. Természetesen megjegyzést is fűzhetünk a könyvekhez: hány kötet, hol található stb.

1. lista

```
10 REM KONYVTARNYILVANTARTAS
20 DIM A$(300,5)
30 POKE 53280,5
40 POKE 53281,7
45 PRINT"■"
50 PRINTTAB(11);"BB  SOFTWARE   1987"
60 PRINT"■■■
```

```
0 REM KONYVTAR-MEGNYITO
10 FOR I=1 TO 6
20 X$="@:D"+STR$(I)+",S,W"
25 Y$=LEFT$(X$,3)+RIGHT$(X$,5)
30 OPEN 2,8,2,Y$
40 PRINT#2,0
50 CLOSE 2
60 X$="@:K"+STR$(I)+",S,W"
65 Y$=LEFT$(X$,3)+RIGHT$(X$,5)
70 OPEN 2,8,2,Y$
80 PRINT#2,CHR$(13)
90 CLOSE 2
100 NEXT
```

2. lista

A program kezelése igen egyszerű, mivel működés közben mindenről tájékoztatást kapunk. Kiválaszthatunk egy könyvet bármelyik információjának megadásával, ide értve a megjegyzést is. A könyveket témakörök szerint csoportosítottuk, tehát más-más nyilvántartási egységbe kerül például egy regény és egy lexikon.

Ha a nyilvántartást meg akarjuk tekinteni, először az „információ"-t kell kérnünk, majd a témakör megadása után választhatunk: megadjuk valamely információját, vagy minden könyv nyilvántartására kíváncsiak vagyunk. A „lapozás"-t bármelyik billentyű lenyomása kiváltja. Adatbeírás esetén az új könyv felvételén kívül a régieket is módosíthatjuk. Ilyenkor figyeljük a sorszámot, mert a módosítás ennek alapján lehetséges.

Természetesen a program kapacitása nem végtelen. Témakörönként legfeljebb 300 könyv adatát tárolhatjuk, de a program csekély változtatásával a témakörök száma növelhető.

Első indítás előtt futtassuk le a *2. listán* közölt megnyitóprogramot, különben nyilvántartásunk hibajelzéssel leáll. Ezt a programot minden egyes új lemez megnyitásakor is le kell futtatnunk.

Valószínűleg sokaknak okozott már bosszúságot, hogy egy kölcsönadott könyv nem került elő. Nos, ez a probléma a program rendszeres használatával, a megjegyzés rovat segítségével megoldható.

BÁRTFAI BARNABÁS

Titkosírás

C64, VIC–20

Az alábbiakban egy titkosírási módszert, ennek számítógépes megvalósítását és programját mutatjuk be. A programot a BICSKE 1 néven alakult számítógépes társaság készítette. A módszer lehetővé teszi elemi adatok kódolását, dekódolását. Kis átalakítással lemezen akár egész adatfájlokat titkosíthatunk. Könnyen megoldható a nyomtatóhoz illesztés is. Külön feladat az ilyen titkosírás megfejtése program segítségével.

Az eljárás a következő. Vegyünk fel egy kulcsszót, például: PETI. Ez lesz a titkosítás alapszava. Ezután keressük ki az ábécé szerinti legelső betűt és írjunk alá egy egyest, a második betű alá kettest és így tovább, egészen az utolsó betűig:

a kulcsszó PETI
3142

Ezután már csak kódolni kell a titkosítani kívánt szöveget: olyan hosszú részekre szabdaljuk, mint a kulcsszó, és a szövegrészletek betűit a kulcsszónak megfelelően olvassuk ki.

Például: MA CSÜTÖRTÖK VAN
3142314231423142

A kódolt szöveg: ACM ÜÖSTTKRÖVN A

A program jelenleg csak 88 karakternyi szöveget képes átfordítani, de némi hozzáértéssel könnyen átalakítható hosszabb szöveg kódolására is. A program egyébként jó példa kezdőknek, mivel a tömbkezelést is jól szemlélteti és sűrűn alkalmazza a ciklusok egybeágazását.

AKLI PÉTER

```
100 REM ----------TITKOSIRAS----------
110 PRINTCHR$(147):PRINT:PRINT:PRINT
120 PRINT TAB(15)"TITKOSIRAS":PRINT
130 PRINT" MAX.88 KARAKTER IRHATO BE."
140 PRINT
150 PRINT" NEM HASZNALHATO A : ES A
160 PRINT:PRINT:PRINT:PRINT
170 PRINT"       USS LE EGY BILLENTYUT!"
180 GET A$:IF A$="" THEN180
190 PRINTCHR$(147)
200 INPUT"                    KOD";KOD$
210 INPUT" (DE)KODOLANDO SZOVEG";SZO$
220 INPUT"KODOLAS=K DEKODOLAS=D";MI$
230 IF MI$="K" OR MI$="D" THEN 250
240 PRINT"";:GOTO 220
250 REM =====KULCSOK ELOALLITASA=====
260 K=LEN(KOD$):H=LEN(SZO$)
270 DIM K1$(K):DIM KO(2,K)
280 FOR I= 1 TO K :K1$(I)=MID$(KO$,I,1)
290 NEXT
300 FOR S=0 TO 2 : FOR O=1 TO K
310 KO(S,O)=O : NEXT O,S
320 FOR I=1 TO K
330 FOR J=1 TO K-1
340 IF K1$(J)>K1$(J+1) THEN GOSUB 560
350 NEXT J,I
360 FOR I=1 TO K
370 FOR J=1 TO K-1
380 IF KO(0,J)>KO(0,J+1) THEN GOSUB 640
390 NEXT J,I
400 REM ==========FOPROGRAM==========
410 IF MI$="K" THEN SV=1
420 IF MI$="D" THEN SV=2
430 GOSUB 700
440 X$="":H1=LEN(SZO$):N=0
450 FOR J=1 TO H1/K
460 FOR I=1 TO K
470 O=KO(SV,I)+K*N
480 X$=X$+MID$(SZO$,O,1)
490 NEXT I
500 N=N+1
510 NEXT J
520 PRINT
530 PRINT"   A (DE)KODOLT SZOVEG:";X$
540 PRINT:PRINT:PRINT
550 END
560 REM ---------RENDEZES 1-----------
570 A$=K1$(J):K1$(J)=K1$(J+1)
580 K1$(J+1)=A$
590 A=KO(0,J):KO(0,J)=KO(0,J+1)
600 KO(0,J+1)=A
610 A=KO(2,J):KO(2,J)=KO(2,J+1)
620 KO(2,J+1)=A
630 RETURN
640 REM ---------RENDEZES 2-----------
650 A=KO(0,J):KO(0,J)=KO(0,J+1)
660 KO(0,J+1)=A
670 A=KO(1,J):KO(1,J)=KO(1,J+1)
680 KO(1,J+1)=A
690 RETURN
700 IF H/K=INT(H/K) THEN RETURN
710 A=K-(H-(INT(H/K)*K))
720 FOR I=1 TO A
730 SZO$=SZO$+CHR$(32):NEXT:RETURN
```

Programvédelem

PLUS/4

Ezzel az ötlettel megvédhetjük saját programunkat az illetéktelen kutakodóktól. Megvalósításához csak egy Turbo Former programra van szükségünk.

Töltsük be gépünkbe saját programunkat. Változtassuk meg úgy, hogy első sorába az alábbi POKE-okat írjuk:

POKE 774,0 a LIST parancs letiltása
POKE 816,136 a SAVE parancs letiltása
POKE 806,103 a RUN/STOP billentyű letiltása
POKE 814,239 a LOAD parancs letiltása

Ha az átírással elkészültünk, mentsük el programunkat a POKE-okkal együtt, majd töltsük be a Turbo Formert és mellé saját, átírt programunkat, amelyet az előbb mentettünk ki. Ezután a turbó segítségével mentsük ki futtatás nélkül programunkat a SAVE"programnév",7,1 utasítással. A programot most már nem lehet leállítani a RUN/STOP billentyűvel, és ki sem tudjuk menteni, sem kilistáztatni. A POKE utasításokat a következőképpen lehet érvényteleníteni:

POKE 774,110 LIST engedélyezése
POKE 816,164 SAVE engedélyezése
POKE 814,74 LOAD engedélyezése
POKE 806,101 RUN/STOP engedélyezése

Ez a módszer persze csak egy a sok közül. Bárki kidolgozhat újabbakat is programjai védelmére.

BÁCSI PÉTER

BASIC és gépi kód

Legutóbb a DEF USR utasítást feldolgozó rutinnal kapcsolatosan megismertük a betöltőprogramnak egy olyan változatát, melynek segítségével tetszés szerinti szabad memóriaterületre tölthetünk egy gépi kódú rutint. A DEF USR kipróbálásához egy kis táblázatot állítottam össze, melynek segítségével a BASIC beépített függvényeinek egy részét USR-függvényként használhatjuk. Említettem, hogy a C16-os HEX$ és DEC függvénye a másik két géptípuson hiányzik. Most ezeknek a C64-es változatát ismerjük meg. Megírásukhoz felhasználtam a C16 ROM-listáinak idevágó részeit, de néhány apró részletben eltértem azoktól.

HEX$

A disassemblált rutin az *1. listán* látható. Működését — az eddigiekhez hasonlóan — az utasításcímekre hivatkozva ismertetem.

$CØB2: a CHKNUM szubrutint hívjuk, mely ellenőrzi, hogy a függvény numerikus-e. Ha nem, akkor TYPE MISMATCH hibaüzenettel leáll a program.

$CØB5: a korábban már megismert GETADR rutin a FAC tartalmát címformátumúvá alakítva a $14—15 címekre teszi le.

$CØB8—CØBA: az A regiszterbe töltjük az eredménysztring hosszát, majd hívjuk azt az eljárást, mely a sztringnek lefoglalja a helyet, majd a címét a $62—63, hosszát a $61 címre teszi.

$CØBD—CØC6: az indexregiszter nullázása után a címformátumú egész szám magasabb, majd alacsonyabb helyiértékű bájtját az A regiszterbe töltve meghívjuk azt a szubrutinunkat, mely a regiszter tartalmát binárisról kétbájtos hexadecimális ASCII karakterlánccá konvertálva írja az előbb lefoglalt sztringterületre.

$CØC9: befejezésül arra a ROM-rutinra ugrunk, mely a létrehozott karakterlánc jellemzőit a sztringverembe viszi a további feldolgozásra.

Ezután a binárisról hexadecimális karakterekké alakítható eljárás következik.

$CØCC—CØD1: az A regiszter tartalmát a verembe mentjük, majd a magasabb helyiértékű 4 bitet az alacsonyabb helyiértékűek helyére tolva szubrutinként hívjuk azt az eljárást, mely ezt a fél bájtot hexadecimális karakterré alakítja, és a sztring Y-nal indexelt pozíciójába írja.

$CØD4: a veremből visszatöltjük az A regiszter korábbi tartalmát és „rácsorgunk" az imént szubrutinként hívott eljárásra.

$CØD5—CØDD: az A regiszter alsó fél bájtját hexadecimális karakterré alakító rutin. Az AND utasítás a felső fél bájtot nullázza. Ha ezután a regiszterben 10-nél kisebb szám marad, $3Ø-at hozzáadva megkapjuk a szám ASCII megfelelőjét. 10 vagy nagyobb szám esetén további hetet hozzá kell adni, hogy a kívánt karaktert kapjuk. A $CØDB címen levő ADC operandusában csak 6-ot látunk, de a C-bit értéke itt csak 1 lehet, ami szintén hozzáadódik az A regiszter tartalmához.

$CØDF—CØE2: az indirekt indexelt utasítással letesszük az A regiszter tartalmát a kívánt helyre, majd növeljük az indexregiszter tartalmát, végül visszatérünk a hívás helyére.

```
c0b2 20 8d ad    jsr $ad8d
c0b5 20 f7 b7    jsr $b7f7
c0b8 a9 04       lda #$04
c0ba 20 7d b4    jsr $b47d
c0bd a0 00       ldy #$00
c0bf a5 15       lda $15
c0c1 20 cc c0    jsr $c0cc
c0c4 a5 14       lda $14
c0c6 20 cc c0    jsr $c0cc
c0c9 4c f9 b6    jmp $b6f9
c0cc 48          pha
c0cd 4a          lsr
c0ce 4a          lsr
c0cf 4a          lsr
c0d0 4a          lsr
c0d1 20 d5 c0    jsr $c0d5
c0d4 68          pla
c0d5 29 0f       and #$0f
c0d7 c9 0a       cmp #$0a
c0d9 90 02       bcc $c0dd
c0db 69 06       adc #$06
c0dd 69 30       adc #$30
c0df 91 62       sta ($62),y
c0e1 c8          iny
c0e2 60          rts
c0e3 ea          nop
```

1. lista

```
c0e4 20 82 b7    jsr $b782
c0e7 c9 05       cmp #$05
c0e9 b0 3b       bcs $c126
c0eb 85 24       sta $24
c0ed a0 00       ldy #$00
c0ef 84 62       sty $62
c0f1 84 63       sty $63
c0f3 c4 24       cpy $24
c0f5 f0 27       beq $c11e
c0f7 b1 22       lda ($22),
c0f9 c9 30       cmp #$30
c0fb 90 29       bcc $c126
c0fd c9 3a       cmp #$3a
c0ff 90 0a       bcc $c10b
c101 c9 41       cmp #$41
c103 90 21       bcc $c126
c105 c9 47       cmp #$47
c107 b0 1d       bcs $c126
c109 e9 07       sbc #$07
```

2. lista

$CØE3: a NOP utasítás nem tartozik az eljáráshoz, szerepe a soron következő DEC rutintól való elválasztás. Én nemigen használom erre a célra, de nyomtatásban megjelent programokban gyakran találkozom vele: * nb"basic—25.4"

DEC

A DEC-nek megfelelő USR-függvény működése is eltér egy kissé C16-beli párjától: nem fogad el szóközöket a paraméterként megadott karakterlánc kifejezésben. Úgy vélem, hogy ez a lehetőség senkinek sem hiányzik.

A *2. listán* láthatjuk a disassemblált változatot. Megfigyelhetjük, hogy az előforduló abszolút címhivatkozások csak ROM-rutinokra utalnak, saját belső címre nem. Ez azt jelenti, hogy nincs szükség az előző részben megismert különleges betöltési módszerre, a rutin anélkül is bármely szabad helyre betölthető. Mi most nem használjuk ki ezt a lehetőséget.

Az eljárás működése:

$CØE4: azt a ROM-rutint hívjuk, mely beolvassa a sztringparaméter adatait. A cím $22—23-ra, a hossz az A regiszterbe kerül.

$C0E7—C0EB: ha a paraméter hosszabb, mint 4 karakter, a hibarutinra ugrunk, különben a hosszt $24-en tároljuk.

$C0ED: Y-t és a munkaterületet ($62—63) nullázzuk. Y-t a karakterláncon belüli pozíció címzésére használjuk majd indexregiszterként.

```
c10b e9 2f      sbc #$2f
c10d 0a         asl
c10e 0a         asl
c10f 0a         asl
c110 0a         asl
c111 a2 04      ldx #$04
c113 0a         asl
c114 26 63      rol $63
c116 26 62      rol $62
c118 ca         dex
c119 d0 f8      bne $c113
c11b c8         iny
c11c d0 d5      bne $c0f3
c11e 86 0d      stx $0d
c120 a2 90      ldx #$90
c122 38         sec
c123 4c 49 bc   jmp $bc49
c126 4c 48 b2   jmp $b248
```

$C0F3—C0F5: megvizsgáljuk, hogy Y elérte-e a sztring hosszát. Ha igen, a befejező részre ugrunk.

$C0F7—C107: a paraméter következő karakterét az A regiszterbe töltjük, és megvizsgáljuk, hogy érvényes hexadecimális karakter-e. Érvénytelen karakter esetén a hibarutinnal folytatjuk.

$C109—C10B: számjegy esetén $30-at levonunk az A regiszter tartalmából, betű esetén még további hetet. Itt is figyelembe vettük a $C0DB-nél leírtakat, természetesen az ADC helyett az SBC jellemzőivel.

$C10D—C110: az A regiszter tartalmát 4 bittel balra toljuk, hogy az alacsonyabb helyiértékű fél bájt a magasabb helyére kerüljön.

$C111—C119: X-et ciklusszámlálóként használva az A regiszter tartalmát a C-biten keresztül a munkaterületre toljuk. Figyeljük meg, hogy itt a bájtok nem a címformátumnak megfelelő sorrendben következnek.

$C11B—C11C: az Y regiszter tartalmának növelése után feltétlen ugrás következik arra az utasításra, mely azt vizsgálja, hogy Y tartalma elérte-e a $24-en tárolt hosszt. Y-ban itt nem lehet nulla.

$C11E—C123: itt X-ben nulla van, ezt töltjük a típusjelzőbe, ami azt jelenti, hogy az eredmény numerikus. A következő utasítások előkészítik a $62—63-on tárolt előjel nélküli egész szám lebegőpontossá való átalakítását. Ezt az átalakítást a $BC49-en kezdődő interpreter rutin végzi el.

$C126: hibarutin, közvetlen ugrás az "ILLEGAL QUANTITY ERROR"-ra.

A betöltőprogram

A két új rutin közös betöltőprogramja a *3. listán* található. A DATA sorok számozása most már folyamatosan követi az előzőeket, hiszen megszűnt a betöltési címmel való összefüggés. Begépelésnél fel lehet használni az előző részben közölt C64-es változat programját, változás a DATA sorokon kívül a 110, 140, 220 és 270 számú sorokban van. A mostani kezdőcím úgy lett meghatározva, hogy az új rutinok az előző részben leírt program után legyenek betöltve; ezen értelemszerűen lehet változtatni.

Betöltéskor ügyeljünk arra, hogy az USRDEF program még ne legyen aktív. Ezért célszerű, ha előbb a mostani programot futtatjuk le és csak utána az előzőt. Ennek okára alább adok magyarázatot.

Az új függvények a már korábban leírt módon DEF USR utasítással definiálhatók. A HEX$-nek megfelelő függvény címe 49330, a DEC-nek megfelelőé 49380. Ha a betöltőprogramban a KC értékét megváltoztatjuk, a fenti címek is változnak: az előbbi a KC programban megadott értéke, az utóbbi 50-nel több. Vigyázzunk, hogy a definiálásnál ne a KC változót, hanem annak a programba beírt konstans értékét használjuk.

Figyelem! Hiba!

A most közölt betöltőprogram tesztelése közben fedeztem fel az USRDEF eljárásnak egy nagy hibáját: nem szereti az FN-függvényeket. A velük való találkozásnál befagy a rendszer. Ez az oka a betöltőprogrammal kapcsolatos — fentebb említett — bonyodalmaknak. A hiba okát egyelőre nem ismerem pontosan, de remélem legközelebb már tájékoztatást tudok adni róla. Addig is azt tanácsolom, hogy ha az USRDEF eljárást használjuk, kerüljük az FN-függvényeket.

BARNA LÁSZLÓ

```
100 rem hex$ + dec     c64
110 z=256 : kc=49330
120 k=kc : print chr$(147)
130 def fnd(x)=x-48+7*(x>64)
140 for l=1 to 10
150 s=0 : read a$,b
160 print chr$(19);peek(63)+z*peek(64);
170 for i=1 to len(a$) step 2
180 a=16*fnd(asc(mid$(a$,i,1)))+fnd(asc(
mid$(a$,i+1,1)))
190 poke k,a : s=s+a : k=k+1 : next
200 if b<>s then print "hiba!" : stop
210 next
220 for i=1 to 3
230 read d,a : a=kc+a : b=int(a/z)
240 poke kc+d,a-z*b
250 poke kc+d+1,b
260 next
270 end
280 data 208dad20f7b7a904207db4a0,1478
290 data 00a515201ac0a514201ac04c,947
300 data f9b6484a4a4a4a2023c06829,1203
310 data 0fc90a9002690669309162c8,1079
320 data 60ea2082b7c905b03b8524a0,1445
330 data 0084628463c424f027b122c9,1384
340 data 309029c93a900ac9419021c9,1290
350 data 47b01de907e92f0a0a0a0aa2,998
360 data 040a26632662cad0f8c8d0d5,1566
370 data 860da290384c49bc4c48b2,1172
380 data 16,26,21,26,32,35
```

3. lista

A Precomp Plus előfordító program

Plus/4, VC—15XX

Az előfordító a Plus-Comp Plus/4-es BASIC fordítóprogram-rendszer új tagja. A Plus-Comp fejlesztése során az első feladat egy BASIC programok fordítására alkalmas fordítóprogram kidolgozása volt. A fejlesztés mindig kompromisszumra kényszeríti a programozót, amit csak később tud feloldani. Ezt a lépéshátrányt szünteti meg a Precomp Plus.

A fejlesztés célja az utasításkorlátok feloldása volt. Ezenkívül kiváltotta a kezdetben nehézkes direktivakezelést oly módon, hogy a forrás BASIC program előfordítása után a Plus-Comp fordítás közben helyesen kezeli a BASIC 2.0-ás verzióban nem értelmezett utasításokat, függvényeket is.

Az előfordító működésének vizsgálata segítséget nyújthat olyan keresztfordító vagy konvertáló rendszerek létrehozásához, amelyek különböző gépeken használt BASIC programok átalakítására vagy fordítására alkalmasak (például C64 Simon's BASIC→Plus/4, BASIC 3.5, C64 BASIC 2.0→TVC IS BASIC 1.2 stb.).

A B-G-S Comp Commodore 64-es fordítórendszer Simon's BASIC és Supergraphik 64 moduljainak előfordítókkal való kiterjesztése már fejlesztés alatt áll. Ezzel 98 százalékos kompatibilitást lehet majd elérni.

Hasonló elven fog működni a T—68 fordítórendszer is, amely C64-es BASIC és gépi kódú programokat alakít majd át Plus/4-en futtatható állapotúra (játékprogramok átalakítására nem lesz alkalmas).

Az előfordító előnyei

Korlátozások feloldása

A Plus-Comp V—1.0 nem tudta az alábbi utasításokat végrehajtani:

- = DO UNTIL/WHILE *
- = LOOP UNTIL/WHILE *
- = RESTORE (n) *
- = ELSE *
- = TRAP RESUME
- = TRON TROFF

A *-gal jelölt utasítások BASIC 2.0-ás verzióra redukált alakba kerülnek előfordítás után, és így a fordító optimalizált alakban tud kódot generálni.

A hibakezelést elősegítő és nyomkövető utasítások feloldását a Plus-Comp V-2.0-ás verziója végzi majd el.

Automatizált direktivakezelés

Ha a lefordítandó kifejezés tartalmazott olyan utasítást vagy függvényt, amely nem volt értelmezve a CBM BASIC 2.0-ban, a kifejezés elé direktivát kellett helyezni, ha az utasítás nem a kifejezés első tagja volt. Ezt a kényelmetlenséget szüntette meg az automatikus direktiva-generálás és a különlegesen kezelendő GRAPHIC utasítás vizsgálata.

Sebességnövekedés

A RESTORE (n) utasítás kivételével a transzformált utasítások végrehajtása jelentősen meggyorsult a BASIC-hez viszonyítva (DO—LOOP ciklus esetén hatszoros lett).

```
0 rem"       ********************
1 rem"       *   Precomp  +     *
5 rem"       * Plus-comp  v-1.0 *
6 rem"       * BASIC elofordito *
7 rem"       * BCS  1987.XI.VI  *
8 rem"       ********************
10 :
30 color0,1,2:color4,5,5:Print"■":rem " * szinbeallitas
40 c$="                          :rem " * munka $ helyfoglalasa
41 cc%=0                         :rem " * belovasott BASIC utasitassor hossza
50 c$="                         " :rem " * munka $ feltoltese chr$(32)-vel
51 c$=c$+c$+c$+c$+c$+c$+c$+c$+c$+c$+c$+c$+c$+"
60 u$=c$                         :rem " * torlo $ = 255 db space
61 h4$=""                        :rem " * generalt utasitas gyujtoje
62 dim da%(500),dP%(500),d0%(500),l0%(500),hu$(500)
63 rem                                  " * tomb dimenzionalasok =====
64 rem                                  " * da%( = DATA sorszam - 32767
65 rem                                  " * dP%( = DATA elemszam mutato
66 rem                                  " * d0%( = DO   sorszam - 32767
67 rem                                  " * l0%( = LOOP sorszam - 32767
68 rem                                  " * hu$( = hibauzenetek
69 :
70 ki$="■■■
```

Az előfordító működése

Fordítási menetek

Az előfordító egy vagy két menetben fordít. A fordítási menetek száma attól függ, hogy talált-e olyan utasítást az előfordító, amelynek transzformálásához az egész programon egyszer végig kell mennie (RESTORE (n), DO—LOOP, EXIT).

A Precomp Plus fő részei

a: INIT (inicializáló rész)
b: GET—B—LINE (egy BASIC sor munka$-ba olvasása)
c: SCANNER (elemző első és második menet)
d: GENERATOR (új kifejezés generálása)

INIT. A program által használt mutatókat állítja be, tömbfoglalást és munkasztring-helyfoglalást végez. Beállítja a fordítási üzenetek kijelzési szintjét, megnyitja olvasásra az előfordítandó programfájlt.

GET—B—LINE. Ez a rövid gépi kódú program a munka$-ba a lemezről beolvas egy BASIC sort. Az elemző ezt a sztringet vizsgálja.

SCANNER. Az elemző a munka$-ban lévő BASIC utasítássort vizsgálja bájtonként. A felismert kulcsszavak alapján végzi a további elemzést és a kódgeneráló ennek alapján dolgozik.

GENERATOR. A felismert kulcsszavak és kifejezések alapján készíti a továbbfordításra alkalmas lemezes programfájlt.

A hibavizsgálat elemzés közben történik, csak a Precomp Plus megkövetelte formátumok alapján. A Plus-Comp által fel nem derített szintaktikus hibák kiszűrését a Precomp Plus V—1.1 szintaktikai modulja végzi majd el.

INIT Ø000—Ø380

(lásd a listát)

0—90-es sorok. Az inicializáló rész elvégzi a rendszermutatók és a táblázatok foglalását. A legelső sztringváltozónak kiemelkedő jelentősége van (C$), mivel a GET—B—LINE program ebbe a sztringbe olvas be egy sort a lemezről. Az elemző a memóriából olvassa ki a soron következő bájtot, amelynek kezdőcíme a sztringváltozó sztringtároló mutatója. A beolvasott sor feldolgozása után az US-gel történik a munkasztring törlése (255 db szóköz). A CC% a beolvasott BASIC sor hosszát tartalmazza. A H4S segédváltozó, amelyhez a generált utasítás bájtonként lesz hozzáírva.

A táblázatok részletes ismertetésére később, az egyes utasítások elemzésénél visszatérünk.

95—260-as sorok. A program az 1. képernyővel *(1. ábra)* jelentkezik be. Az elemzési szintek a felismert utasítások képernyőn való megjelenítését vezérlik.

Ø: ezen a szinten minden üzenet kiíródik. Ilyenek a kihagyott utasítások (TRAP RESUME, TRON TROFF), a lecserélt utasítások (ELSE, RESTORE (n), DO UNTIL, WHILE, LOOP, EXIT), a kötelezően direktívába helyezendő kifejezések és a Precomp által szintaktikusnak minősített hibák (GRAPHIC paraméter . . .).

1: itt a lecserélt és a hibás utasítások,

2: itt pedig csak a hibák jelennek meg a képernyőn és a nyomtatón.

A kijelzési szintek választhatósága a képernyőn való sorszám + üzenet követését segíti elő.

Az üzenetek három típusát — 0-ás szinten — a 2. képernyőkép *(2. ábra)* mutatja be. Az „átlépett" utasítások előtt nem áll semmilyen karakter, a lecserélt utasításokat a balra mutató nyíl, a hibásakat az inverz alak jelöli.

A fordítandó program nevének megadása előtt lehetőség van a katalógus átnézésére.

270—330-as sorok. Az első menet alatt létrehozott előfordított fájl 1. prgnév néven generálódik. Az olvasásra megnyitott programfájl első két bájtja a töltési címet tartalmazza (ahová a BASIC program a memóriába töltődik = $1001). Ezek kiolvasása után következik a BASIC program olyan formában, ahogyan a tárba töltődik. A láncolási címeknek az előfordító szempontjából nincs jelentőségük.

340—350-es sorok. A Plus-Comppal grafikus üzemmódban lefordított programot SYS-szel indítva a GETKEY utasításnál a programfutás ILLEGAL DIRECT ERROR hibaüzenettel leáll a RUNMOD mutató hibás tartalma miatt. Ezért a generált program 0-ás sorába a következő utasításokat írjuk be:

Ø POKE 129,128:REM > > B

A REM > > B a Plus-Compot vezérli úgy, hogy fordítás közben csak a hibaüzenetek jelennek meg.

360—380-as sorok. A képernyőre írja a fejlécet.

BÁRÓ CSABA

(Folytatjuk)

```
      ********************
      * PRECOMP  PLUS    *
      *  PRECOMPILER     *
      *   PLUS-COMP      *
      *     V-1.Ø        *
      *      '87         *
      ********************


ELEMZES KIJELZESI SZINTJE1 <Ø-2>
--------------------------------

 : Ø :  OSSZES UZENET KIJELZESE
 : 1 :  KICSERELT UTASITASOK + HIBAK
 : 2 :  PRECOMP SYNTAX HIBAK

KERI A KATALOGUST IGEN / NEM ?

PROGRAM NEVE  : ? TESZT
```

1. ábra

```
1: 230        PRECOMP PLUS V1.0

              PROGRAM : TESZT

---------------------------------------

  10   :      TRON
  20   :      TROFF
  30   :      FUGGVENY
  40   :      FUGGVENY
  50   :    ← ELSE
  55   :      FUGGVENY
  60   :      GETKEY
  70   :    ← GRAPHIC
  80   :    ← GRAPHIC
 120   :      RESUME
 130   :      TRAP
 170   :      GRAPHIC PARAMETER 1
 180   :      GRAPHIC PARAMETER 1
 190   :      GRAPHIC PARAMETER 2
 210   :      FUGGVENY
 220   :      FUGGVENY
 220   :      FUGGVENY
* 230   :      TRAP
```

2. ábra

Forgató szubrutinok 6502-re

A Personal Computer Worldben (1986. október, 203. oldal) Andrew Johnson három igen szép forgató rutint jelentetett meg, amelyek hiányoztak a 6502 (6510, 7510, 8502) mikroprocesszor-család utasításkészletéből. Nézzük meg közelebbről, hogy mit is tudnak ezek a rutinok, és ha lehetséges, próbáljuk meggyorsítani a működésüket.

Az első rutin egy, a Z80-as mikroprocesszoron meglévő aritmetikai jobbra fordulást szimulál (*1. ábra*).

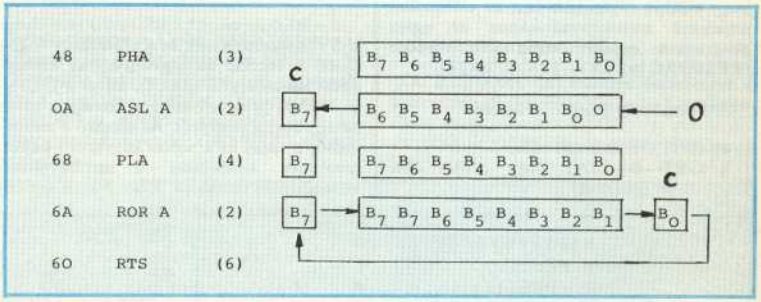
Nézzük végig sorról sorra részletesen, hogy mi is történik itt. Az első sorban a veremtárolóba tesszük az akkumulátor tartalmát, majd pedig a hetedik bitet (B_7) beforgatjuk a C-be (a carry flagbe, az átvitelt jelző bitbe). A harmadik sorban újra előveszszük az eredeti számot a veremtárolóból, majd a negyedik sorban a ROR A hatására a B_7 bitet újra beírjuk a hetedik bitbe, a carry flagbe pedig az eredeti 0. bit — a B_0 — kerül. Így gyakorlatilag a B_7 bitet újrageneráltuk: az a helyén is maradt és egy helyiértékkel jobbra is átmásolódott, emiatt ezt a rutint ASR-nek (Arithmetic Shift Right-nak), azaz aritmetikai jobbra forgatásnak lehet nevezni.

Zárójelben a végrehajtáshoz szükséges ciklusidőket tüntettük fel. Így összesen 17 óraciklusra van szükség a rutin végrehajtásához. A C64 esetében ez egyúttal 17 µs-ot jelent. (Az ábrán az akkumulátor és a carry flag tartalmát tüntettük fel bitekre bontva.)

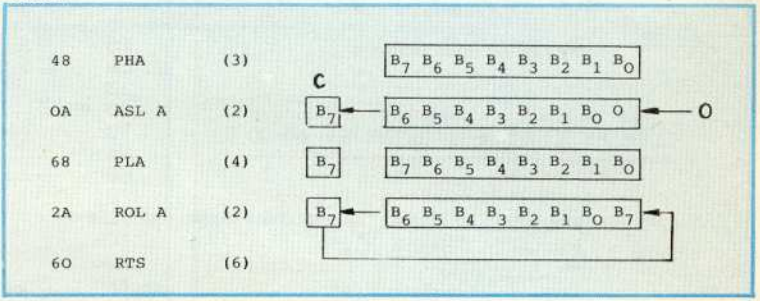
A második forgató rutin szintén hiányzik a 6502 utasításkészletéből. Ez egy 8 bites körbeforgatást valósít meg balra, a carry flagen keresztül (*2. ábra*).

Mint a korábbi forgató rutinnál, itt is először a veremtárolóba mentjük ki az akkumulátor (.A) értékét, majd az ASL A utasítással a C-be forgatjuk bele a hetedik bitet. (Itt ASL A helyett állhatna ROL A is, mivel ennek az utasításnak egyedüli célja, hogy a hetedik bitet, a B_7-et a carry flagbe, az átvitelt jelző bitbe másolja át!) A lényeg itt is a harmadik utasításnál található, hiszen itt állítjuk vissza a korábban elrontott biteket azzal, hogy a veremtárolóból újra elővesszük az eredeti bájtot. A ROL A utasítással már csak be kell fejezni a megkezdett műveletet, és mint az ábráról is látható, az eredeti hetedik bit a C-ből beíródik a nulladik bit helyére, azaz így egy nyolcbites balra forgatás valósul meg. Ez a rutin is öt bájt hosszúságú és 17 óraciklus szükséges a végrehajtásához.

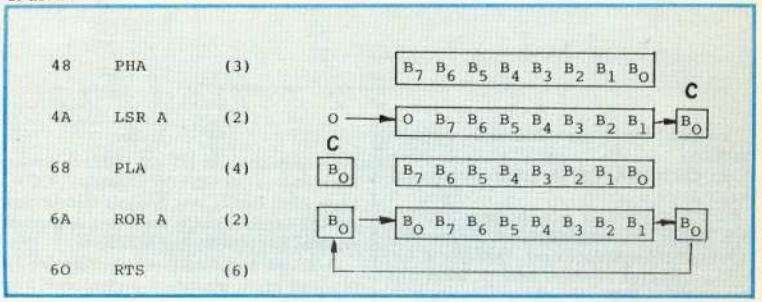
A harmadik rutin az előzőhöz hasonló módon egy nyolcbites jobbra forgatást valósít meg (*3. ábra*). Itt a nulladik bitet másoljuk be a C-be az LSR A utasítással (LSR A helyett itt is állhatna ROR A), utána visszaírjuk a veremtárolóból az akkumulátor eredeti tartalmát, majd most már a C-n keresztül egy jobbra forgatást valósítunk meg a ROR A utasítás segítségével, aminek hatására a korábbi nulladik bit beíródik a carry flagből a hetedik bitpozícióba. Ezzel pedig létrejött a nyolcbites — teljes értékű — jobbra forgatás, hiszen nem vesztettük el



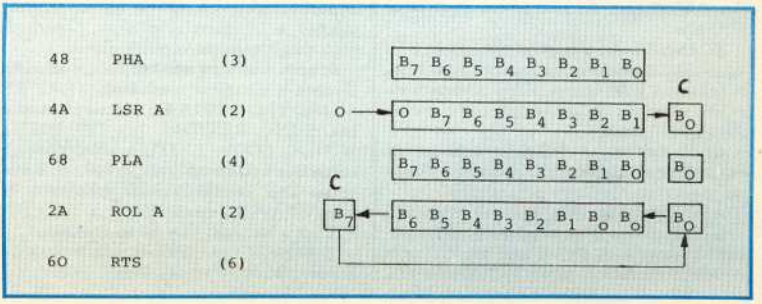
1. ábra



2. ábra



3. ábra
4. ábra



közben egyik bitet sem. A 6502 mikroprocesszor-család utasításkészletéből ez a fajta utasítás is hiányzott eddig.

Az előzőekhez hasonlóan a negyedik fajta forgatást is felépíthetjük magunknak (*4. ábra*). Itt a legelső forgatáshoz hasonlóan egy olyan balra forgatásról van szó, amikor a nulladik bit a helyén is marad és át is másolódik a mellette lévő bitpozícióba, a fenti rajznak megfelelően. Ehhez a forgató rutinhoz nem fűzünk bővebb magyarázatot: a rutin működése az előzőekben elmondottak és a bájt bitenkénti forgatását bemutató rajz alapján megérthető.

Nézzük meg ezek után, hogy hogyan lehetne meggyorsítani ezeknek a rutinoknak a működését. Kézenfekvőnek látszik az a gondolat, hogy rövidíteni kellene a veremtárolóba történő beíráson és visszaolvasáson: különösen a veremtárolóból történő visszaolvasás tart hosszú ideig. Ezen úgy rövidíthetünk, ha a PHA-t helyettesítjük TAX-szel. Így az akkumulátor tartalmát az .X tárolóba mentjük ki, ez a korábbi három gépi ciklus helyett két gépi ciklust jelent. Ehhez hasonlóan a korábbi PLA-t a TXA-val kell ilyenkor helyettesítenünk, tehát az eredeti .A értékét az .X tárolóból kell visszaírnunk újra az akkumulátorba. Ez a művelet a korábbi négy óraciklus helyett kettő alatt hajtható végre. Így a veremmel kapcsolatos műveletek elmaradnak, de ennek a gyorsaságnak az az ára, hogy az .A-n kívül a rutinjaink használják az .X tárolót is, és emiatt annak korábbi tartalmát elrontják. Ezzel a megszorítással sikerült lerövidítenünk a rutinok működését a korábbi 17 óraciklusról 14-re. (Természetesen az .X tároló helyett választhattuk volna az .Y tárolót is: ekkor TXA és TAX helyett TYA-t, illetve TAY-t kellett volna írnunk.)

Az első két forgató rutint — ahol a hetedik bitet kell a carry flagbe, az átvitelt jelző bitbe átmásolni — másképpen is jelentősen meggyorsíthatjuk. Ezt a megoldást Robert Gardner-Medwin javasolta (PCW, 1987. március, 176. oldal). A megoldás lényege, hogy egy összehasonlítást végzünk. Vegyük például az első, a hetedik bitet jobbra forgatás mellett újrageneráló programot:

```
C9 80  CMP # 128      (2)
6A     ROR A          (2)
60     RTS            (6)
```

Az összehasonlítás az akkumulátor tartalmát nem rontja el, mindössze beállítja az N, Z és C biteket, az összehasonlítás eredményétől függően. Így például, ha az .A tartalma kisebb 128-nál, akkor a C flag törölt állapotú lesz (C=Ø). Amennyiben az .A≥128 feltétel teljesül, akkor az átvitelt jelző bit egyenlő lesz 1-gyel (C=1). Ez egyúttal azt is jelenti, hogy ennek az összehasonlításnak a hatására az akkumulátor hetedik bitje átmásolódik a C-be. Vegyük itt azt észre, hogy az összehasonlítást egy speciális számmal végeztük, amely bináris felírásban a következő: %10000000.

Így a rutint a korábbi öt bájtról négyre sikerült lerövidíteni, és a végrehajtási idő is jelentősen lerövidült: a korábbi 17, illetve 14 óraciklusról 10 óraciklusra!

Ez az előbbi módszer még alkalmazható a második forgató rutinnál is, amely egy teljes nyolcbites körbe forgatást valósít meg balra:

```
C9 80  CMP # 128      (2)
2A     ROL A          (2)
60     RTS            (2)
```

Itt is a 128-cal történő összehasonlítás hatására másolódik át a hetedik bit a C-be, majd onnan a ROL A hatására a nulladik bitbe. (Ez is négy bájt hosszúságú művelet, amely 10 óraciklus alatt hajtható végre.)

Ez az előbbi módszer már nem alkalmazható a nulladik bit átmásolására a carry flagbe, az átvitelt jelző bitbe, mert nem tudjuk úgy megválasztani az összehasonlítás alapját, hogy az összehasonlítás hatására a nulladik bit átmásolódjon a C-be. Ebben az esetben tehát a gyorsítás számára marad a már korábban ismertetett lehetőség.

Felmerülhetne még bennünk, hogy próbálkozzunk meg a BIT művelet felhasználásával, amely szintén átmásolja a hetedik és a hatodik bitpozícióban lévő biteket, de sajnos — a mi szempontunkból — az átmásolás nem jó helyre történik, mivel a C-t a BIT művelet nem érinti; ez a művelet a B_7-et N-be (a negatív számot jelző bitbe), B_6-ot pedig V-be (a belső túlcsordulást jelző bitbe) másolja át. Így ezeket az átmásolt biteket nem tudjuk a forgatásra felhasználni, hiszen a forgatásban érdekelt összes művelet — ASL A, LSR A, ROL A, ROR A — mind a carry flag, az átvitelt jelző bit felhasználásával zajlik.

SZABÓ PÉTER PÁL

Propaganda Spectrummal

Nyilvános adatbázisok

Egy holland számítógép-amatőr tudósít arról, hogy egy kis 48 kbájtos Spectrummal, modem segítségével több száz nyilvános adatbázishoz tud hozzáférni.

A nyilvános adatbázisokból nem csupán érdekes dolgokat tudhatunk meg, hanem többségük közvetítésével a hozzám hasonlók „nyilvános leveleket" is írhatnak. Egyes adatbázisok kapacitása több ezer képernyőoldal. Ezeket a tulajdonos cégek maguk nem is győzik kitölteni. Mégis azt szeretnék, hogy adatbázisuk minél több emberhez szóljon, tehát a lehető legtöbb oldalt akarják megtölteni érdekes, változatos témákkal.

Egyszer lapozgatva az egyik rendszert, olvasom, a tulajdonos — egy kereskedelmi vállalat — munkatársakat, úgynevezett információszállítókat keres, hogy érdekes információkhoz juttassa olvasóit. A Spectrumomban rögtön előkészítettem a választ, amit közleményként elküldtem a rendszer „postaládájának", személyesen a rendszer vezetőjének címezve. Azt írtam, kész vagyok néhány oldalt összeállítani az eszperantó nyelvről. Már másnap csöngött a telefonom s nekem szegezték a kérdést: hány oldalt kívánok? Szerényen azt válaszoltam, hogy mondjuk ötöt. „Annyit nem érdemes, ötven vagy akár száz oldalt is biztosítunk önnek!" — hangzott a válasz. A beszélgetés végén azonnal megkaptam egy olyan belépési kódot, amely ötven oldal erejéig írásra jogosított.

Íme a büntetés dicsekvésemért: nekiállhattam egy olvasásra érdemes szöveget összeállítani az eszperantóról, s azt oldalról oldalra továbbítani az adatbázisoa, Rotterdamba. Rengeteg szakirodalmat bújtam át, újságok, szórólapok százait, míg végül megszületett harminc képernyőoldal. Felosztottam fejezetekre, amelyek közül az olvasók az első oldalon levő menü segítségével választhatnak.

Az egyik fejezet címe: Hogyan hangzik az eszperantó? A legnagyobb csodálkozásomra kiderült, hogy ez a cím tette kíváncsivá a legtöbb olvasót, hogy megnézze a hozzá tartozó szöveget. Az emberek egyszerűen az eszperantó kiejtéséről akartak hallani. De honnan ismerem az egyes képernyőoldalak olvasottságát? Nos, az adatbázisnak van egy rendszerstatisztikája, amely pontosan követi, hogy melyik oldalt hányszor választották ki. Ez figyelmen kívül hagyja az adott oldal információ szállítójának a belépési kódját, tehát én magam nem tudom a saját oldalaim statisztikáját meghamisítani gyakori beolvasgatásokkal. Az információ szállító egyébként csak a saját részének statisztikáját jogosult megtekinteni.

Amint elhelyeztem az első harminc oldalamat a rendszerben, elkészítettem egy speciális oldalt, hogy felhívjam a figyelmet az eszperantó rovatra. Ezt az oldalt elküldtem több tucat másik adatbázisba. A szövege a következő: „Bonan tagon!" (Jó napot kívánok!) — s a többi holland nyelven. „Vagy ön még nem beszél eszperantóul? Akkor kérjen információt (következik a hollandiai eszperantó intézmények felsorolása) vagy nézze meg a V. I. S. adatbázist a 4350. oldaltól, telefonszám 010—4332035".

Az egyik, Hollandia egész más részén levő adatbázisból azonnal megkaptam a dörgedelmes választ: „Miért nézzék a MI OLVASÓINK a V. I. S. adatbázist? Szíveskedjen gondoskodni arról, hogy a mi rendszerünkben is legyen ilyen rovat! Erre a célra szívesen rendelkezésre bocsátunk önnek ötven oldalt!"

Ez volt az a reagálás, amire egyáltalán nem számítottam. Természetesen azonnal oda is átküldtem a már elkészített harminc oldalamat. Ezután már két rendszerben egészítgettem ki az eszperantó rovatomat, hozzátéve újabb oldalakat, frissítve a már meglevő információt.

Mindez 1987 októberében történt. December közepére már négyre növekedett azoknak a rendszereknek a száma, amelyek forgalmazzák az eszperantóról szóló oldalaimat, együttvéve 150 képernyőoldal terjedelemben. Valamennyi éjjelnappal olvasható. Íme a teljes lista:

Az adatbázis neve	Telefonszáma	Kezdő oldal
V. I. S.	010—4332035	4350
CGSV	01883—12475	2001
Willem Tell	038—542699	12
Kameleon	05202—19295	2750

Az oldalak karbantartása természetesen elég sok pénzembe kerül az interurbán telefonköltségek miatt. Végül ismét hangsúlyozni szeretném, hogy nem vagyok számítástechnikai szakember, csupán egy egyszerű amatőr.

Gerard van der HORST

Mit tud a C nyelv? II.

Mutatók (pointerek)

Nem újdonság, hogy a magas szintű nyelvekben megengedik olyan változók definiálását és használatát, amelyek más változók címét tartalmazzák. A C nyelvben is igen alaposan és precízen átgondolták a mutatók hatékony, logikus szerepét. Ez kényelmessé és biztonságossá teszi az indirekt címzést.

Lássunk akkor egy mutatót! Tegyük fel először, hogy egy sztringet akarunk megcímezni. Definiáljunk tehát egy karakter típusú sztringet és egy mutatót!

```
char mess [ ] = „Ez egy üzenet!\n";
char *cpnt;
```

Egy mutatót ugyanúgy kell deklarálni, mint egy közönséges változót, de neve előtt csillaggal jelezzük, hogy ez a változó egy mutató. Hogyan adhatunk értéket a mutatónak és hogy használhatjuk fel azt?

```
cpnt = &mess[0];
```

Az „&" („és") operátor segítségével képezhetjük bármelyik objektumunk címét.

Érdekességként megemlítjük, hogy egy tömb neve az első tömbelem címével egyenértékű, tehát a fenti értékadást így is el lehetne végezni: cpnt = mess;

Most a cpnt mutató a mess nevű üzenet első karakterére (E) mutat. Ha most indirekt címzéssel kiolvassuk ezt a karaktert, akkor könnyen ki is nyomtathatjuk a putchar() könyvtári függvény segítségével:

```
putchar(*cpnt);
```

Itt a cpnt elé írt csillag jelzi az indirekciót — vagyis, hogy nem a cpnt tartalmát, hanem az általa címzett karakteres változó tartalmát kívánjuk felhasználni. Hogyan férhetünk most hozzá a következő karakterhez? Nos, ennek címe éppen eggyel nagyobb, mint az előző karakteré — adjunk tehát egyet a mutatóhoz:

```
cpnt = cpnt + 1;
```

Ezután az újabb putchar() hívás már a második karaktert fogja kiírni és így tovább.

Mivel egy változó értékének megnövelése vagy csökkentése igen gyakori művelet, erre külön operátort vezettek be a C megalkotói, a ++ és −− operátorokat. A nyelv egyik erénye, hogy e két operátort ún. prefix (tehát változó előtti) és postfix (változó utáni) helyzetben is használhatjuk. Legyen „a" és „b" két egész típusú változó. Ekkor:

```
int a,b;
    a = 5;
    b = a++;    (*„b" 5, „a"
                 6 lesz *)
    b = ++a;    (*„b" 7, „a"
                 is 7 lesz *)
```

Tehát a különbség az, hogy a postfix pozícióban felhasználás után, prefix pozícióban felhasználás előtt növekszik meg a változó értéke. Ezek az operátorok minden változó esetében használhatók, tehát mutatókra is alkalmazhatjuk őket:

```
cpnt++; ill. ++cpnt;
```

Az előbbi első lépésben felhasználja a mutató értékét, majd megnöveli, az utóbbi pedig először növeli, utána használja fel. Ennek számos esetben vehetjük hasznát. Tegyük fel például, hogy egy sztringet át akarunk másolni egy másik sztringre. A pnt1 mutató címezi a kiindulási sztringet, a pnt2 pedig a célsztringet. Ekkor a

```
*pnt2++ = *pnt1++;
```

utasítás átmásolja a soron következő karaktert, egyúttal megnöveli a mutatókat — készen állunk tehát a következő karakter másolására, csak valahogy meg kell ismételtetnünk az utasítás végrehajtását.

A mutatók használatában a C nyelv éppen itt, a címaritmetika területén hozott újat. Mi történik, ha nem karakterek, hanem például lebegőpontos változók tömbjét címezzük meg egy mutatóval? Ekkor a ++ operátor természetesen nem eggyel, hanem néggyel, azaz egy lebegőpontos változó hosszával növeli meg a mutatót — így az a tömb soron következő elemére fog mutatni.

Megengedett az, hogy mutatókhoz egész kifejezéseket adjunk, sőt, két azonos típusú mutatót ki is vonhatunk egymásból. Az eredmény az indexek különbsége lesz:

```
float farr [20];
float *pnt1, *pnt2:
int a;
        pnt1 = &farr[8]; pnt2
        = &farr[17];
        a = pnt2 − pnt1;
```

Ekkor „a" értéke nem 36 lesz, amennyi a címek közötti fizikai különbség, hanem 9 — a megcímzett elemek indexeinek különbsége.

Természetesen definiálhatunk olyan mutatókat is, amelyek nem egyszerű változókra vagy egyszerű változókból képzett tömbökre, hanem struktúrákra mutatnak:

```
struct flyobj fly;          (*struktúra
                             definíció*)
struct flyobj *flypnt;      (*mutató
                             struktúrára*)
```

Ez a mutató egy flyobj típusú struktúra címét fogja tartalmazni. Most — értéket adva a mutatónak — közvetlenül felhasználhatjuk azt a megcímzett struktúra tagjainak elérésére:

```
flypnt = &fly;      (*mutató −
                     előkészítés*)
flypnt → code       (*egy tag
= „*";               felülírása*)
flypnt → posx       posx + flypnt
= flypnt →          →stepx;
```

Egy mutató típusa, mint láttuk, a címaritmetikai műveletek elvégzésében játszik szerepet. Ha viszont a mutató struktúrára címez, akkor a fordító azt is ellenőrzi, hogy helyesen hivatkoztunk-e a struktúra tagjaira, azaz a → operátor után szereplő szimbólum valóban struktúratag-e és valóban ennek a struktúrának tagja-e. Tehát igen precízen kell hivatkoznunk a struktúra tagjaira, és nem fenyeget az a veszély, hogy a mutatók segítségével egészen más struktúrák elemeit fogjuk megcímezni.

A struktúramutatók használatának két tény ad különleges fontosságot. Az egyik az a természetes dolog, hogy struktúrákból tömböt is szervezhetünk:

```
struct flyobj flyarr[10];
```

Ekkor a tömb elemeit igen kényelmesen címezhetjük meg egy azonos típusú mutató segítségével.

A másik, ennél fontosabb tény, hogy struktúrát „egészben" nem tudunk függvénynek paraméterként átadni, bár ez igen sokszor kívánatos lenne. Ilyen esetben segítenek a mutatók: ha struktúrát nem is adhatunk meg paraméterként, a struktúra címét egyszerűen átpasszolhatjuk a tömbök címéhez hasonlóan.

Ciklusszervező utasítások

A C nyelv igen egyszerű és kényelmes ciklusszervező utasításokkal rendelkezik. Mi ezek közül csak egyet mutatunk be, hogy megismerjük a C egy ragyogóan használható sajátosságát.

A while utasítás a while kulcsszóból, egy kifejezésből és egy utasításból áll:

```
while (kifejezés)
{
   utasítás(ok);
}
```

A kifejezés tetszőleges C kifejezés lehet. A program először kiértékeli ezt a kifejezést, majd ha igaz, azaz értéke különbözik

a 0-tól, végrehajtja az utasítás(oka)t, és visszatér a ciklus elejére; újra kiértékeli a kifejezést, és ha igaz . . . Amennyiben a kifejezés értéke hamis (azaz 0), akkor a program végrehajtása a blokk után folytatódik.

Lássunk egy nagyon egyszerű példát:

```
int a;
    a = 20;
    while (a)
    {
        a - - ; (* a csökkentése *)
    }
```

Ez a kis ciklus éppen húsz alkalommal fut le. Sok értelme azonban nincs. A következő legyen egy értelmesebb példa. Tegyük fel, hogy a pnt1 mutató egy 0 karakterrel lezárt karakteres tömb kezdetére mutat, a pnt2 mutató pedig egy olyan memóriaterületre, amelyre az előbbi karaktertömböt át akarjuk másolni. Lássuk a ciklust először „gyalogosan", majd pedig egy elegánsabb, C-szerűbb változatban:

```
while (* pnt1 != 0)
{
  *pnt2++ = * pnt1++;
}
```

A while alatti kifejezés most már egy igazi logikai kifejezés — a program összehasonlítja a != (nem egyenlő) operátor két oldalán szereplő kifejezéseket, az egész kifejezésnek pedig igaz az értéke, ha a közéjük írt feltétel teljesül. Tehát a fenti ciklus pontosan addig ismétlődik, amíg a kiindulási tömböt címző pnt1 nem a tömböt lezáró 0 kódú karakterre mutat. A ciklusmag pedig egy fentebb már ismertetett szerkezet: először használja, majd módosítja (növeli) a mutatók értékét — tehát először átmásolja a pnt1 által címzett karaktert a pnt2 által címzett memóriapozícióra, majd megnöveli pnt1 és pnt2 értékét.

Ennek a programrészletnek van egy roppant súlyos hibája: a tömböt !ezáró 0 kódú karaktert nem másolja át, így a céltömb „végtelen" hosszú lesz, hacsak külön utasítással a végére nem másoljuk a záró zérót.

Lássuk most az igazi, a valóban C-szerű megoldást! Azt használjuk ki, hogy egy egyszerű értékadás is egy kifejezés, melynek van saját értéke:

```
int a, b, c, d;
    a = b = c = d = 0;
```

Először d kap egy 0 értéket, de a d = 0 értékadás is egy kifejezés, melynek értéke az, amit d kapott — most a 0. Tehát ez a 0 „öröklődik" előbb a c, majd a b, végül pedig az a változóba. Ha pedig ez így van, akkor egy értékadást nyugodtan beleírhatunk a ciklus vezérlő részébe is:

```
while ((*pnt2++ = *pnt1++) !=0)
{
}
```

Mit csinál ez a (C nyelvben teljesen szokványos) utasítássorozat? Először kiértékeli a while alatti kifejezést, amely egy logikai kifejezés. Jobb oldala 0, bal oldala egy értékadás, amelyet egy külön zárójel emel ki. A program tehát az értékadást végzi el először. Ez a szokásos módon megy végbe, megnöveli a mutatókat, de kimenő értéke az, ami a pnt2 által címzett memóriapozícióba íródott — tehát az átmásolt karakter. A program tehát az átmásolt értéket veti össze a nullával — így a ciklus addig ismétlődik, míg csak át nem másolja a kiindulási tömböt lezáró 0 karaktert is, hiszen addig egyetlen karakterkódja sem lehet 0! A ciklus magja ez esetben üres, hiszen minden szükséges tevékenységet elvégzett már a ciklus vezérlő része.

Többszintű mutatók

A mutatók egyik legrokonszenvesebb sajátossága az, hogy nemcsak alaptípusokra vagy struktúrákra tudnak mutatni, hanem mutatókra is. Egy mutatót címző mutató tehát egy változó címének címét tartalmazza. Lássunk erre is példát. Tegyük fel, hogy valamilyen célból ki akarjuk nyomtatni a hónapok nevét. Ekkor persze a neveket előre elhelyezzük egy-egy karakteres tömbben:

```
char jan[ ] = „Január"
char feb[ ] = „Február"
char mar[ ] = „Március"
.
.
.
char dec[ ] = „December"
```

Definiáljunk most egy újabb tömböt, amely karaktereket címző mutatókból áll:

```
char *month = {jan, feb, mar, . . . , dec};
```

Ez utóbbi definíció egy olyan mutatókból álló, tizenkét elemű tömböt definiál, amelynek elemei rendre megcímzik a jan[ ], feb[ ], mar[ ], . . . dec[ ] karaktertömböket. Így tehát például month[5] a hatodik mutató, amely — minthogy 0-tól kezdődik az elemek számozása — a jun[ ] nevű tömb kezdőcímét tartalmazza. Próbáljunk most meg egy olyan mutatót definiálni, amely a month[ ] tömb tagjainak megcímzésére képes:

```
char **monthpnt;
```

azaz mutató egy karakter(tömb) címét tartalmazó mutatóra. Lássuk most az értékadást:

```
monthpnt = month;
```

mivel month egy tömb, amelynek neve egyenértékű a kezdőcímével. Ekkor:

```
*monthpnt
```

a month tömb első eleme, vagyis egy olyan mutató, amely a jan[ ] tömbre mutat:

```
**monthpnt
```

pedig egyenértékű a jan[ ] tömb első elemével, azaz a J karakterrel! Ha pedig rendelkezésünkre áll a kívánt hónap sorszáma (például 6. hónap), akkor e hónap nevét megcímzi egyfelől a

```
month[5]
```

másfelől pedig a

```
*(monthpnt+5)
```

hivatkozás is.

Ha elég elszántak vagyunk, természetesen tovább is léphetünk. Ha az a szándékunk, hogy a hónapok nevét több nyelven is kiírjuk, akkor minden hónap nevét előkészítjük a kívánt nyelveken, definiálunk egy mutatókból álló tömböt, amelynek elemei rendre címzik a neveket tartalmazó sztringet, majd definiálunk egy újabb tömböt, amely mutatókra címző mutatókból áll:

```
char jan_eng[ ]    = „January";
char feb_eng[ ]    = „February";
.
.
.
char dec_eng[ ]    = „December";
                   (*angol
                   hónapnevek*)
char               = {jan_eng,
*month_eng[ ]      feb_eng, . . .
                   dec_eng}
                   (*mutatótömb
                   a címzésükhöz*)
char jan_frn[ ]    = „Januar";
char feb_frn[ ]    = „Février";
.
.
.
char dec_frn[ ]    = „Decembre";
                   (*francia
                   hónapnevek*)
char               = {jan_frn,
*month_frn[ ]      feb_frn, . . .
                   dec_frn}
                   (*mutatótömb
                   a címzésükhöz*)
```

Eddig semmi újdonság nincs. De lássuk a mutatók tömbjeinek címét tartalmazó mutatók tömbjét:

```
char **monthpa[ ] = {month,
month_eng, month_frn};
```

és egy, e tömb elemeinek megcímzését lehetővé tevő mutatót:

```
char ***monthp;
```

amely, mint látni fogjuk, karakterek címét tartalmazó mutatók címét tartalmazó mutatók megcímzésére alkalmas (bocsánat!):

```
monthp = monthpa; (*monthpa egy tömb*)
```

Ekkor, minthogy maga monthp a monthpa[ ] tömb címe, *monthp a monthpa tömb első eleme, az pedig mutató a month című tömbre.

**monthp a month tömb első eleme, tehát egy mutató a (magyar) jan[ ] tömbre. Végezetül pedig

***monthp

nem más, mint a jan[ ] tömb első eleme, egy karakter, mégpedig a példa szerint egy J karakter.

Természetesen nem szükségszerű, hogy egy mutató tömbelemre mutasson, így tehát az sem, hogy egy „kétcsillagos", tehát mutató címét tartalmazó mutató egy mutatókból álló tömböt címezzen meg. Többszörös indirekciót azonban a leggyakoribb esetben mégis ilyen szituációkban használunk.

A fenti kis példák és a hozzájuk tartozó eszmefuttatások talán jól körvonalazzák azokat a szokatlan lehetőségeket, amelyek oly használhatóvá teszik a C nyelvet rendszer- vagy rendszerközeli programok írására. Hiszen az előbbihez hasonló, háromszorosan indirekt hivatkozás még egy „mindenre elszánt" assembler programozót is arra késztetne, hogy alapos megfontolás után lásson csak neki a program megvalósításának. Egy percig sem állíthatjuk azt, hogy ilyen szerkezetek megvalósítása a C nyelvben nagyon egyszerű lenne: a mutatók kezelésének jól átgondolt filozófiája azonban lényegesen egyszerűbbé teszi ilyen és ehhez hasonló rémtettek elkövetését.

PETHŐ ÁDÁM

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.

Szuperchipek, kártyaújdonságok

A Western Digital céget itthon csak a szakemberek ismerik, és ők is mint a hajlékony- és merevlemezegységek illesztésére szolgáló integrált áramkörök gyártóját. Valóban, a lemezegységeket illesztő áramköröket az IBM PC megjelenése előtt szinte minden gyártó ettől a cégtől vásárolta.

Az IBM a hajlékonylemezegység-illesztőt máshonnan szerezte be, így a Western Digital cég ezen a területen hátrébb szorult, de a merevlemezeknél megmaradt a világelsősége, sőt a teljes kártyáknál is megszerezte ezt. Megtartva profilját, néhány éve létrehozta Paradise Systems nevű leányvállalatát más integrált áramkörök, teljes kártyák gyártására (például az elmúlt évben 1,5 millió video integrált áramkört gyártott, ami az IBM után a legnagyobb mennyiség). A Paradise most egy hasonlóan sikeresnek ígérkező területen jelentkezett.

1. táblázat

IBM VGA üzemmódok
Grafikus

Üzemmód	Felbontás	Kártyatípus	Színválaszték
4,5	320 × 200	VGA	4 256 k közül
		MCGA	4 25 k közül
		EGA	4 64 k közül
		CGA	4
6	640 × 200	VGA	2 256 k közül
		MCGA	2 256 k közül
		EGA	2 64 k közül
		CGA	2
D	320 × 200	VGA	16 256 k közül
		EGA	16 64 k közül
E	640 × 200	VGA	16 256 k közül
		EGA	16 64 k közül
F	640 × 350	VGA	monokrom
		EGA	monokrom
10	640 × 350	VGA	16 256 k közül
		EGA	16 64 k közül
11	640 × 480	VGA	2 256 k közül
		MCGA	2 256 k közül
12	640 × 480	VGA	16 256 k közül
13	320 × 200	VGA	256 256 k közül
		MCGA	256 256 k közül

Szöveges

Üzemmód	Felbontás	Kártyatípus	Oszlop	Sor	Karaktermátrix	Színek
0,1	360 × 400	VGA	25	40	9 × 16	16 256 k
	320 × 400	MCGA	25	40	8 × 16	16 256 k
	320 × 350	EGA	25	40	8 × 14	16 64 k
	320 × 200	CGA	25	40	8 × 8	16
2,3	720 × 400	VGA	25	80	9 × 16	16 256 k
	640 × 400	MCGA	25	80	8 × 16	16 256 k
	640 × 350	EGA	25	80	8 × 14	16 64 k
	640 × 200	CGA	25	80	8 × 8	16
7	720 × 400	VGA	25	80	9 × 16	monokrom
	720 × 350	EGA	25	80	9 × 14	monokrom
	720 × 350	MDA	25	80	9 × 14	monokrom

Az IBM a legújabb PS/2 sorozat a korábbi PC-sorozattól (és az utánzatoktól) az új buszszerkezettel, analóg monitorok használatával, a 3,5" méretű lemezegységekkel és a *beépített* video kijelző csatolóval — Video Graphics Array (VGA) — különbözött. Ez utóbbi csak a sorozat Model 50 és nagyobb sorszámú tagjainál használatos. A Model 25 és 30 ennek egy kevesebbet tudó változatával, a Multi-Color Graphics Array (MCGA) csatolóval van ellátva.

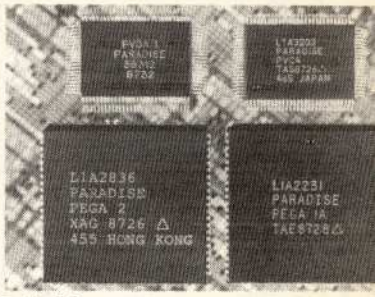
Mit tud ez az új — várhatóan szabványnyá váló — rendszer? A különböző üzemmódokat az *1. táblázatban* foglaltuk össze. Látható, hogy mind a felbontás, mind a színek száma nőtt a korábbiakhoz képest. Várhatóan az új szoftverrendszerek már építeni fognak ezekre a megnövekedett lehetőségekre.

Mit tehetnek a korábbi termékek (elsősorban a másolatok) gyártói? Mivel a 14 VGA-mód elérése a PS/2 beépített szoftverén (BIOS) keresztül történik, a teljes kompatibilitás elérése így lehetetlennek látszik (hasonlóan a PC BIOS problémájához). A hardverben megvalósított megoldás szoftverszimulációja használhatatlanul lassúvá teszi a rendszert. Az egyetlen elfogadható megoldás a gépbe szerelhető kártya szintű hardver/szoftver kombináció, megfelelő áron. Az igazán célszerű megoldás az, ami a korábbi rendszerekkel is kompatibilis marad, mivel ezekre épülnek a korábbi és a jelenlegi szoftverek. (Ezek a Color Graphics Adapter — CGA —, Monochrome Display Adapter — MDA —, Enhanced Graphics Adapter — EGA — és a Hercules Monochrome Graphics.)

A PVGA1 IC

A Paradise cég kezdett a világon először egy VGA, CGA, MDA és Hercules hardver kompatibilis IC-t gyártani, ami szoftver kompatibilis az EGA, MCGA rendszerekkel. Ez az *1. fényképen* látható PVGA1. (A képen a további három IC az áttervezett és két korábbi változatú EGA illesztő.) Az alkatrész a 80286, 80386, PS/2 Model 25 és 30 rendszerek alkalmazási lehetőségeit terjeszti ki a leírt módon, lehetővé téve ezekhez megfelelő kártyák gyártását.

A PC rendszereknél szerzett tapasztalatok szerint a másolatok gyártói gyakran túllépnek az IBM rendszerek lehetőségein. Így született meg például a Hercules grafika, az EGA. Ennek a leendő követelménynek a Paradise elébement, amikor az IC-be olyan üzemmódokat is beépített, amelyek a PS/2 sorozatból hiányoznak *(2. táblázat)*. Ugyancsak jelentős előny a mintegy kétszeres működési sebesség, amit a hatásosabb belső szerkezet — egy-egy 8 és 16 bites adatbuszillesztő, 8 vagy 16 bites speciális BIOS — és a 8 és 16 bites tárakhoz való il-



1. fénykép

2. táblázat

PVGA1 az IBM VGA felett
Kompatibilitás

Üzemmód	IBM	Paradise
VGA	hardver és BIOS	hardver és BIOS
MCGA	BIOS	BIOS
EGA	BIOS	BIOS
CGA	hardver és BIOS	hardver és BIOS
MDA	hardver és BIOS	hardver és BIOS
Hercules	nincs	hardver és BIOS

Jellemzők
Grafikus

Felbontás	Színválaszték	IBM	Színek Paradise	Adatbuszszélesség IBM[1]	Adatbuszszélesség Paradise
320 × 200	262 144	256	256	8 v.16	8 és 16
640 × 480	262 144	16	16	8 v.16	8 és 16
640 × 480	262 144	nincs	256	nincs	8 és 16
800 × 600	262 144	nincs	16	nincs	8 és 16
960 × 720	monokrom	nincs	monokrom	nincs	8 és 16
1024 × 768	monokrom	nincs	monokrom	nincs	8 és 16
1280 × 1024	monokrom	nincs	monokrom	nincs	8 és 16

Szöveges

Oszlop	Sor	IBM	Paradise
80	25	igen	igen
132	25	nincs	igen
132	43	nincs	igen
132	55	nincs	igen

[1] A PS/2 Display Adapter a Model 25 és 30 gépnél 8 bites, a PS/2 alapkártya a Model 50—80-as gépnél 16 bites.

lesztés biztosít. További sebességnövelő tényező a nagyobb órafrekvencia (itt 40 MHz, a PS/2-nél 28). Az IC 1,5 mikron sűrűségű CMOS technológiával készült, ezért egyrészt kis fogyasztású, másrészt igen kis méretű. Ez utóbbit igazolja a *2. fénykép*, amelyen egy IBM VGA illesztővel együtt látható.

A VGA IC-k piacát az ACUMEN Inc. erősen növekvőnek véli: 1987-re 1,2 millió darabot várt, 1988-ra 2,5, 1989-re 3,1 millió darabra számít.

VGA Professional Card

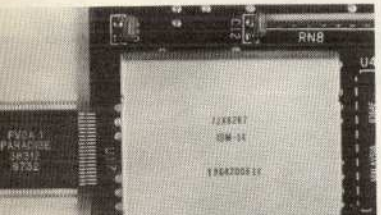
A felhasználók többsége nem kíván kártyát tervezni egy IC-hez, hanem kész kártyákat akar gépébe építeni. A kártyahasználók egy része szeretné kihasználni az IC lehetőségeit, azokat is, amelyek túllépik a VGA-szintet. Ezek számára készült a *3. fényképen* bemutatott kártya.

A kártya a *3. táblázatban* látható monitorokkal dolgozhat. Minden olyan alkalmazásnál használható, ahol fontos a nagy felbontású grafika, a sokkarakteres szövegsor, a nagy színspektrum. Speciális szoftverhajtókat is beépítettek a kártya tárjába a *4. táblázatban* felsorolt szoftverekhez. Ezekkel például a Windows működési sebessége lényegesen megnő.

VGA Plus Card

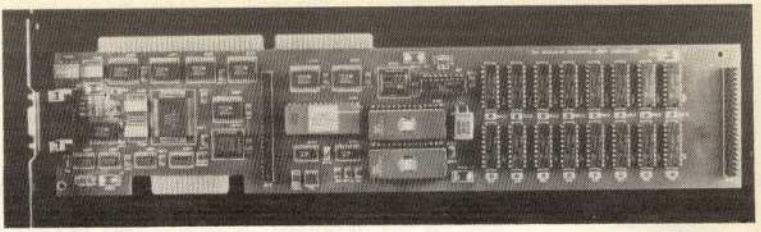
A *4. fényképen* látható kártya azok számára készült, akik gépüknek VGA kompatibilitást kívánnak biztosítani. Ez a kártya a 8514 kivételével ugyanazokhoz használható, mint az előző, olyanhoz is, mint például a NEC Multisync Plus.

A két kártya és az IBM PS/2 Display Adapter összehasonlítása látható az *5. táblázatban*. Lényeges a szakmai jellemzőkön kívül az ár, ami az Egyesült Államokban az ún. ajánlott ár (amit a gyártó a kereskedőknek ajánl eladási árként, és amelynél a kiskereskedelmi ár általában alacsonyabb). Ennek alapján az első kártya azonos árért többet tud, a második pedig hasonló teljesítmény mellett olcsóbb.



2. fénykép

A Western Digital nem hagyott fel eddigi profiljával, sőt — ahogy a cég elnöke, Roger W. Johnson a COMDEX/FALL '87-en nyilatkozta — az új IBM-sorozatnál is meg akarja tartani vezető pozícióját. Ennek érdekében minden fontosabb sorozattag fontosabb kártyáinak továbbépítésében részt vett. Mivel?



3. fénykép

Western Digital IC-k a Model 25/30-hoz

Egy leányvállalatuk dolgozta ki. Az FE2011 típusú IC kezeli a billentyűzetet, egeret, a tárakat (64 k—1 Mbit tartományban). A Model 30-cal regiszter szinten kompatibilis és kb. 25 százalékkal nagyobb teljesítményű.

VGA Professional Card
Monitorválaszték

Monitortípus	Üzemmódok
IBM: 8514	mind 1024 × 768-ig
8513	mind 640 × 480-ig
8512	mind 640 × 480-ig
8503	mind 640 × 480-ig
Több frekvenciás	mind 1024 × 768 vagy a frekvenciahatár által megengedettig

3. táblázat

A WD 57C65 nagy sebességű, CMOS lemezhajtó puffer nélkül alkalmas a Model 25/30, 50, 60 kiszolgálására. A WD 16C552 kétcsatornás, aszinkron kommunikációs IC egyaránt jó a PC XT/AT és a PS/2 sorozathoz. Képes két soros be/kimeneti egységet és egy teljes kétirányú párhuzamos egységet egyidejűleg kezelni. Beépített soros/párhuzamos átalakítója is van. Az adás/vételt egy 16 bájtos FIFO (first-in-first-out, azaz amilyen sorrendben érkeznek, ugyanolyan sorrendben távoznak az adatok) puffer segíti.

A PCLK1 a PVGA1 órajel-generátora. Optimalizálja a hozzáférési időket a beépített szoftver segítségével, dinamikusan változtatva az egyes kimenetein megjelenő órajelek frekvenciáit.

A Model 50/60 IC-k

Az FE 5400 négyelemes készlet 20 MHz frekvencián dolgozó CMOS egységekből áll. Tagjai közül az FE 5000 a perifériakezelő és -vezérlő IC. Ez egy DMA vezérlőt, virtuális módot kezelő logikát, Micro Channel időzítő köröket, órajel-generátort, várakozási és rendszerbuszvezérlőt tartalmaz. Az FE 5010 és 5030 a tárkezelést (256 k—4 Mbit) végzi. Az FE 5020 az adat- és címpuffer.

A cég az IC-ken kívül nagy integráltságú elemeket, felületszerelt technológiát, szabványosított méretű és csatlakozású IC-jű kártyákat is gyárt. Ezeket mind a PC, mind a PS/2 sorozat részére gyártják. Az elnök véleménye szerint az első piaca tovább fog nőni, ha lényeges teljesítménynövelő részegységekkel jönnek ki a cégek, és ők is erre törekszenek.

VGA Professional Card
Szoftverhajtók

1024 × 768	
Microsoft Windows	Lotus Symphony
Digital Research GEM	Ashton-Tate Framework II
Ventura Publisher	Lotus 1—2—3
640 × 480 × 256 szín	
Microsoft Windows 2.0	
800 × 600 × 16 szín	
Microsoft Windows	Ventura Publisher
Digital Research GEM	Lotus Symphony
AutoCAD	Ashton—Tate Framework II
Cadvance	Lotus 1—2—3
132 oszlop	
Lotus Symphony	MicroPro WordStar Series
Ashton-Tate Framework II	WordPerfect
Lotus 1—2—3	

4. táblázat

Western Digital kártyák

A WD 93021—XE egy 20 Mbájt kapacitású kártyán megvalósított lemezegységhajtó és -illesztő elektronika, ami közvetlenül a hajtóra szerelhető. A WD 8003E egy Ethernet adapter. A beépített, nagy sebességű 8 k puffer RAM nem igényel DMA csatornát. A kártyán CMOS LAN IC és Ethernet csatlakozó van.

A WD 8003SH az előzőhöz hasonló, csak a StarLAN típushoz, és a beépített 16 k ROM lemezegységmentes használatot biztosít. Így olcsó telefonvonalak is használhatók a hálózatban.

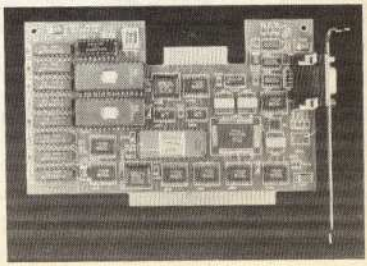
A WD 1006V—MC1 egy ST506 típusú winchester-hajtó a Model 50, 60 és 80 típusokhoz. Teljesen kompatibilis a Micro Channel vonalakkal. Két lemezegységet képes kiszolgálni, maximálisan 2048 hengerrel és 16 fejjel. Az átviteli sebessége 10 Mbit/s.

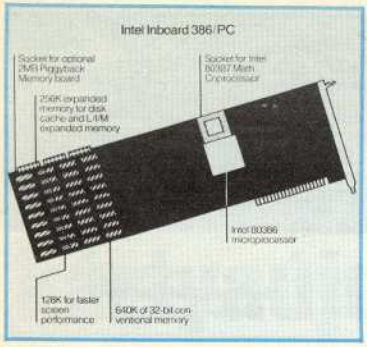
A WD 1007V—MC1 hasonló, csak ESDI illesztőhöz készült, és a maximális sebesség 15 Mbit/s.

A WD 7000—MSC is hasonló, csak SCSI illesztőkhöz való. MSDOS 3.3 és OS/2 kompatibilis.

A WD 8003E/A hasonló a 80003E kártyához, csak a RAM és ROM 64 k.

4. fénykép





1. rajz

WD lemezegységek

A Western Digital új terméke a lemezegység.

A WDPS25/20i a Model 25-be építhető, 21 Mbájt formattált kapacitású lemezegységet és teljes kiszolgáló elektronikát tartalmazó egység. Beépített működést irányító szoftvert is tartalmaz.

A FileCard PS30 egy 32 Mbájtos egy kártyán megvalósított lemezegység, beépített adatkonverterrel és polinom adatkorrekcióval (56 bites).

A WDXT—GEN winchester-illesztőn nincsenek beállítandó átkötések (átkapcsolások), így használata igen egyszerű. A hajtható két lemezegység viszonylag szabadon konfigurálható.

A WD 1002A—27X 56 bites hibakorrekcióval (Error Correction Code-ECC) ellátott winchester-illesztő. Az ún. 2,7 RLL kódolás miatt mintegy 50 százalékkal több adatot tud tárolni, mint az ST506-os típus. A hajtott két lemezegység különböző is lehet. A beépített superBIOS a hibás sávok formattálását is lehetővé teszi.

5. táblázat

Paradise-IBM termékösszehasonlító

Jellemző	VGA Professional	VGA Plus	IBM PS/2 Display Adapter
Ajánlott ár (dollár)	599	399	595
Alak	hosszú	rövid	hosszú
Adatbusz	16 bites	8 bites	8 bites
Üzemmódok			
VGA	igen	igen	igen
MCGA	igen	igen	igen
EGA	igen	igen	igen
CGA	igen	igen	igen
MDA	igen	igen	igen
Hercules	igen	igen	nincs
132 oszlop	igen	igen	nincs
Nagy felbontású Monokrom			
1024 × 768	igen	nincs	nincs
960 × 720	igen	nincs	nincs
800 × 600	igen	nincs	nincs
Színes			
800 × 600 × 16 szín	igen	nincs	nincs
640 × 480 × 256 szín	igen	nincs	nincs

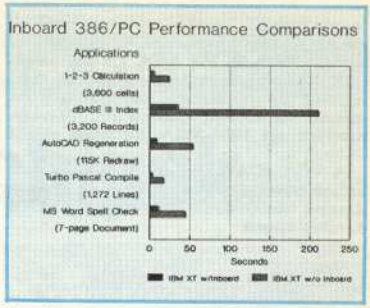
A WD 1003S—WA4 lehetővé teszi hajlékony- és merevlemezegységek, továbbá soros és párhuzamos csatornák egyetlen AT csatlakozósávon történő illesztését. Így egyetlen kártya elég a lemezegységek, nyomtatók és a modem illesztésére. A formátum lehet ST506 vagy ST412, az átviteli sebesség 5 Mbit/s, a fejek száma maximum 16, a hengereké 2048.

A WD 1002A—FOX XT/AT kompatibilis hajlékonylemezegység-illesztő. 1,4 Mbájtig négy egyoldalas vagy két kétoldalas egységet (kazettásat is) kezel. Az egységek felírási sűrűsége és konfigurációja különböző lehet.

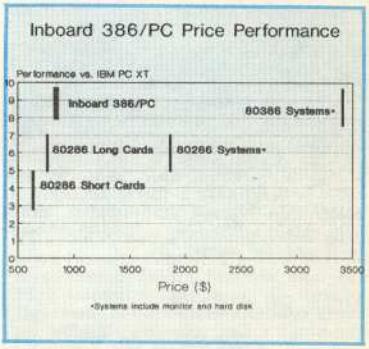
Intel Inboard 386/PC

A lemezegységek egységes szoftverkezelésére a Lotus, az Intel és a Microsoft cég egy LIM rövidítésű, kváziszabványt fogadott el. Ezt valósítja meg az Inboard 386/PC nevű kártya, amelyen 1 Mbájt RAM, 80386 típusú processzor és a beégetett szoftver található *(1. rajz)*. Az eredeti Intel rajzon balról jobbra: foglalat a 2 Mbájt ráhelyezhető tár részére, 256 k tárkiterjesztő a LIM és a diszk gyorstár számára, 128 k képernyőgyorsító, 640 k 32 bites tároló, foglalat a 80387 típusú aritmetikai processzor számára, 80386 típusú mikroprocesszor. A kártya valódi képe látható az *5. fényképen*. A kártya további tartozékok nélkül csatlakoztatható egy XT-hez és alakítja át azt 80386-os géppé.

A csatlakoztatás egy egyszerű kábellel történik, ami a 8088 helyére csatlakozik. A kártya használata által okozott sebességnövekedést mutatja az ugyancsak eredeti Intel készítésű *2. rajz*. (A rajzon a Lotus 1—2—3 3600 cella kiszámítása, a dBASE III 3200 rekordjának indexelése, az AutoCAD 115 k mennyiségű vonal újrarajzolása, a Turbo Pascal 1272 sorának fordítása és a Microsoft Word hétoldalnyi szöveg el-



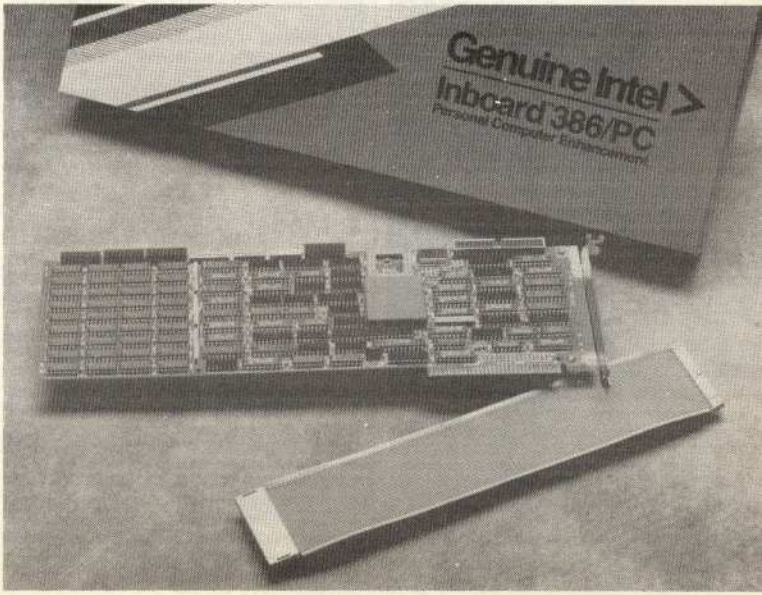
2. rajz
3. rajz



lenőrzése szerepel. A fekete szín a kártyával, a másik anélkül.) Érdekes adatok a *3. rajz* ár/teljesítmény értékei. Itt rövid, hosszú és a vizsgált kártyát, valamint monitort és merevlemezegységet tartalmazó rendszert hasonlítanak össze.

SIMONYI ENDRE

5. fénykép



EGY SZOFTVER, AMELY NÉLKÜLÖZHETETLEN!

MASTER

Oktatóprogram-tervező rendszer
IBM PC XT/AT kompatibilis számítógépre

Ön a rendszer segítségével összeállítja a leckék — és ha szükséges, a tesztek és ellenőrző kérdések — szakmai anyagát, a többit elvégzi ön helyett a MASTER oktatóprogram-tervező rendszer. Segítségével önállóan futtatható oktatóprogramokat készíthetünk. NEM SZÜKSÉGES, hogy az oktatóprogram készítője, használója programozási ismeretekkel rendelkezzék.

Alkalmazási területek:

- Számítástechnikai eszközök, programok, felhasználói rendszerek használatának oktatása, az üzemeltetéshez szükséges ismeretek elsajátításának ellenőrzése.
- On-line help bármilyen felhasználói rendszerhez.
- Termékismertető, árkatalógus.
- Üzemekben, vállalatoknál az új termelő berendezések, eszközök ismertetése. Szakmai továbbképzés.
- Dolgozók rendszeres balesetvédelmi oktatása, vizsgáztatása.
- Valamilyen tanfolyamra jelentkező hallgatók tudásszintjének felmérése, vizsgáztatás.

ÁRA: 77 000—320 000 Ft között, az alkalmazástól függően

Készséggel állunk a MASTER iránt érdeklődő ügyfeleink rendelkezésére további információval és szakmai bemutatóval is. Amennyiben eljuttat hozzánk két db 360 kbyte-ra formattálható floppy lemezt, a tervező rendszer DEMO verzióját ingyen megküldjük önnek tanulmányozás céljából.

„5. G." Számítástechnikai Szolgáltató Kisszövetkezet
1139 Budapest, Kartács utca 27.
Telefon: 295-899, 490-778, 296-446.
Telex: 22-57-46

ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI SZERZŐDÉS = ÖRÖK GARANCIA

FOTOELEKTRONIK ALFA GT NOVOTRADE

COMMODORE, ATARI, TVC stb. személyi számítógépek valamint IBM PC/XT/AT professzionális PC számítógépek és perifériák (floppy, printer) garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.

SZERVIZEINK:

1053 Budapest V., Magyar u. 12–14.	T.: 173-551	Tx: 7621	4034 Debrecen, Holló László u. 14.		
1083 Budapest VIII., Szigony u. 9.	T.: 343-153		5600 Békéscsaba, Bartók Béla u. 37.	T.: 66-27-195	
1191 Budapest XIX., Gábor A. sétány 3.			6726 Szeged, Székelysor 13.	T.: 62-13-377	
3100 Salgótarján, Arany János u. 3.	T.: 32-14-007		7400 Kaposvár, Füredi u. 24.	T.: 82-16-307	
3525 Miskolc, Fazekas u. 1–3.	T.: 46-17-011		7624 Pécs, Jurisics M. u. 17.	T.: 72-11-812	
			9700 Szombathely, Szalonok u. 31.	T.: 94-14-519	

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621
VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.
VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

FELHÍVÁS

A szegedi Tarjánvárosi 4. Sz. Ált. Iskola — az OPI számítástechnikai bázisiskolája — évek óta nyári bentlakásos számítástechnikai táborokat rendez felső tagozatosoknak és középiskolai tanulóknak.
1988-ban a következő témakörökben és turnusokban rendezünk tábort:

Minden turnusban:

- Számítástechnikai alaptanfolyam kezdőknek (ajánlott 5–8. osztályosoknak) (A jelű tanfolyam)
- Számítástechnika haladóknak, alkalmazások (ajánlott 7–8. osztályosoknak, illetve a középiskola 1–2. évfolyamaira járóknak) (B jelű tanfolyam)
  - programozási eljárások, fogások
  - alkalmazói programok
  - fizikai, kémiai, technikai alkalmazások (interfészek, perifériák)
- Számítógépes hardver ismeretek (interfészek tervezése, építése, egyszerű mérő- és szabályozóberendezések építése) (ajánlott 8. osztályosoknak, középiskolásoknak) (C jelű tanfolyam)

Turnusok: (minden turnusban minden témakör szerepel)

június 13—18-ig június 20—25-ig június 27.—július 2-ig július 4—9-ig

Az egyes turnusokat és azon belüli csoportokat a jelentkezők számától függően indítjuk. Minden jelentkező visszajelzést és részletes programot kap. Ekkor küldjük meg a befizetési csekket is.

Jelentkezési határidő: 1988. június 3-ig beérkezően.

A nagy érdeklődésre tekintettel a jelentkezéseket azok beérkezési sorrendjében fogadjuk el.

Jelentkezési cím: Tarjánvárosi 4. Sz. Ált. Iskola, Orkonyi Enikő tanár,
6723 Szeged, Urhajós u. 4. Tel.: 06-62-51-449

Részvételi díj: az A és B jelű tanfolyamokra 1760 Ft
a C jelű tanfolyamokra 2040 Ft

Minden érdeklődőt és jelentkezőt szeretettel várnak:

a Rendezők és az Igazgatóság

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára.
Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi, döntését megalapozza.
A számítástechnikában viszont a széles választékból nem könnyű a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalont.
Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen áttekinthető, kipróbálható, tanulmányozható, összehasonlítható.
Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat.
Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.

PC szalon

Budapest XIII., Sallai Imre u. 6.
315-136, 310-776

Lépjen új korszakba velünk.

NOVOTRADE

Közprogramok

Olvasóink egyre gyakrabban küldenek be olyan hasznos programokat, amelyeket terjedelmi okok miatt nem áll módunkban közölni. Mivel azonban úgy véljük, hogy közérdeklődésre tarthatnak számot, ezeket röviden ismertetjük. Akik használni szeretnék a programokat, a részletes leírást, a programlistákat vagy — ahol erre mód van — a programot adathordozón, levélben megrendelhetik szerkesztőségünktől. A listák, ill. adathordozók másolását a KASZKAD Kisszövetkezet óbudai PÓLUS Szakcsoportja végzi (lásd lapunk 1988/2. számát). A másolási díj listák, leírások esetén oldalanként 8,— Ft, kazetta esetén 150,— Ft, floppy esetén 300,— Ft. Az adathordozóra történő másolás díja az adathordozó árát is magában foglalja.

Közületektől cégszerű megrendelést kérünk, a másolási díjat az MNB 208-42 518-7014 számlára kell befizetni. Magánszemélyek a díjat a KKVMF PÓLUS Szakcsoport, Bp. Bécsi út 94–96. 1034 címre fizethetik be.

A megrendelés az alábbi formában lehetséges:

MIKROSZÁMÍTÓGÉP MAGAZIN SZERKESZTŐSÉGE
1371 Budapest, Pf. 433.

Megrendelem a Mikroszámítógép Magazin 1988/... számában szereplő közprogramok közül az alábbiakat:

A program neve
Csak programlistát és leírást kérek
Példányszám
Csak adathordozót kérek
Mindkettőt kérem

A megrendeléshez csatolom a szolgáltatási díj befizetését igazoló csekkszelvényt. Dátum, név, pontos cím.

A programok ellenőrzése nem áll módunkban, ezért az esetleges programhibákért mi nem vállalhatjuk a felelősséget.

Kérjük, akik saját készítésű programjaikat felajánlják a köz javára, a részletes programleírást, a listát és az alábbiakhoz hasonló rövidített ismertetőt juttassák el címünkre. (Nevük jelenti a garanciát a továbbiakban a megrendelők számára.)

Számkonverter

Programnév: Számkonverzió
Géptípus: TVC
Konfiguráció: alapgép
Adathordozó: kazetta
Terjedelem: 150 BASIC sor
A készítő neve: Paróczay Gábor
Megjegyzés: —
A leírás oldalszáma: 2
A program oldalszáma: 6

A menüvezérelt program decimális/bináris és viszont, logikai 1/0 szintű sorozat/decimális konverziót és ugyanezt csak más megadással, valamint bináris eltolást valósít meg. Egy kiegészítő programmal hexadecimális/decimális konverzió is megy.

Manótervező

Programnév: SPRITE EDITOR 64
Géptípus: C64
Konfiguráció: kazettás vagy lemezegységes háttértároló
Adathordozó: kazetta
Terjedelem: 150 BASIC sor
A készítő neve: Ungváry Levente
Megjegyzés: —
A leírás oldalszáma: 2
A program oldalszáma: 3

A programmal egyszerre 32 manót lehet megtervezni, elforgatni, mozgatni, tükrözni, elmenteni, visszahívni.

Fraktálkészítő

Programnév: SZÓRAKOZTATÓ FRAKTÁL MINTÁK
Géptípus: C64
Konfiguráció: lemezegység
Adathordozó: lemez
Terjedelem: 146 BASIC sor
A készítő neve: Argyelán Péter
Megjegyzés: Simon's BASIC kell
A leírás oldalszáma: 3
A program oldalszáma: 3

A program érdekes (fraktál) alakzatokat állít elő a képernyőn.

ISKOLASZÁMÍTÓGÉP-SZERVIZ
1077 Budapest VII.,
Baross tér 19.
Telefon: 428-999

Vállalja: IBM PC/AT, IBM PC/XT ÉS
Commodore típusú (C16, CPlus/4, C64, C128) gépek javítását,
átalánydíjas szervizét,

egyedi programok, programcsomagok készítését.

DIGITAL
szelvény
Mikroszámítógép Magazin
1988. június

Egy sarokkal olcsóbb!

A DIGITAL Számítástechikai Szaküzlet (1026 Budapest, Szilágyi Erzsébet fasor 35.) Sinclair-termékekre szakosodott: elsősorban a ZX81-es és Spectrum gépekhez használatos eszközöket, programokat árul. De kínál más gépekhez való tartozékokat (például botkormányokat), számítástechnikai alkatrészeket (integrált áramköröket stb.) és zsebszámológépeket is.

Aki ebben az üzletben a lapunkból kivágott sarokszelvényt átadja, vagy megrendelésével együtt oda elküldi, minden hónapban más-más cikket olcsóbban vásárolhat meg. A kedvezmény a szelvényen feltüntetett hónapban érvényes. Minden árengedménnyel vásárolt darabhoz le kell adni egy szelvényt.

A bolt utánvéttel szállítást is vállal, és a szokásos 6 hónap helyett *1 év garanciát ad*

Az e havi kedvezmény

Univerzális nyomtatóillesztő Spectrum gépekhez 5000 Ft, engedmény 10%

Az illesztőnek a doboz oldalán elhelyezett RESET gombja védve van a véletlen megnyomás ellen. Segítségével bármely programot a tápfeszültség kikapcsolása nélkül törölhetünk, így nem kell a tápcsatlakozót kihúzni a gépből.

Az illesztő program nélkül kompatibilis az Interfész 1-gyel.

Bekapcsoláskor az illesztő alapállapotban van. Ilyenkor nincs soremelés (a legtöbb nyomtatóban van egy belső kapcsoló, amellyel az automatikus soremelést be lehet állítani, de ez akkor a programból már nem kapcsolható ki és be) és a nyomtató 80 karaktert gépel soronként. Ezt egy egyszerű POKE 23679, K utasítással tudjuk megváltoztatni, ahol K a kívánt soronkénti karakterszám.

A paranccsal hívható menüből elérhető funkciók az alábbiak.

Nyomtatóhoz illesztés

A választható típusok: Seikosha, GP80, GP100, GP250; Epson MX80, RX80, FX80; Star Micronics típusok és egyéb. Ez utóbbi esetben el kell készíteni és be kell tölteni a nagyfelbontású grafikát előállító programot.

Parancs- vagy szövegválasztás
Soremelés tiltása vagy engedélyezése
Állapotjelzés

A menü utolsó sorában lévő információ megmutatja, hogy az illesztő éppen milyen aktuális állapotban van:

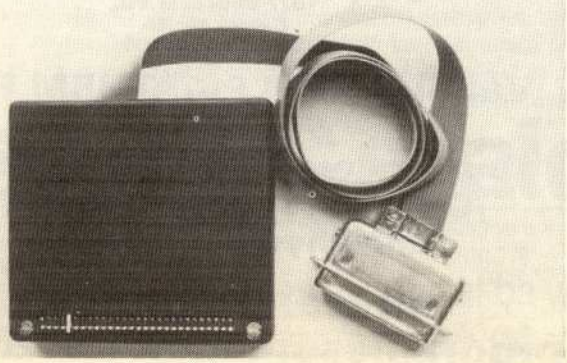
STATUS A BCD E
„A" a nyomtatótípusnak megfelelő szám
„B" a parancs/szöveg állapot
„C" a képernyőállapot
„D" az ESC 0 vagy 1 állapotát mutatja.

Az „E" hiánya az automatikus soremelés hiányát jelzi. Ha itt LF jelet látunk, akkor a soremelés aktualizálva van.

Az állapotok addig vannak érvényben, amíg a tápfeszültséget ki nem kapcsoljuk vagy új utasítással felül nem írjuk.

(Vigyázat! Néhány printerállapotot a nyomtató ki- és újra bekapcsolással ki lehet törölni!)

SE



Hardver

Multiplexerek és demultiplexerek

A multiplexer olyan áramkör, amely a címzésnek megfelelően a bemeneteire kapcsolódó vonalak valamelyikét a kimenetre kapcsolja, azaz a kimeneten megjelenik az adott bemenet logikai állapota.

A demultiplexer ennek a fordítottját teszi: egy bemeneti vonalat a címzésnek megfelelően a kiválasztott kimeneti vonalra kapcsol.

Az *1. ábrán* kapukból felépített, 4 bemenetű multiplexert mutatunk be.

A D kimenet az AØ és A1 címbemenetek állapotának függvényében a DØ...D3 adatbemenetre kapcsolódik. Például ha A/Ø = L, az A1 = H szintű, akkor a D2 bemenet kapcsolódik a kimenetre.

Komparátorok

A komparátorok olyan áramkörök, amelyek két bináris értéket hasonlítanak össze egymással. A három legfontosabb összehasonlítási feltétel: az egyenlőség, a kisebb és a nagyobb viszony. Két bináris érték egyezőségének az a feltétele, hogy minden bitjük megegyezzen. A komparátornak csak akkor van a kimenetén H szint, ha a bemenetén a két szám egyenlő. Ha a két összehasonlítandó szám egybites, akkor „kizáró vagy" (ekvivalencia) kaput használhatunk. N bit esetén az n számú ekvivalenciakapu kimeneteit egy ÉS kapuhoz csatlakoztatva jutunk az N bites komparátor áramkörhöz, ahogy ezt a *2a. ábrán* bemutatjuk. A *2b. ábrán* egy 2×4 bemenetű (4 bites) komparátor elvi rajzát mutatjuk be. Ennek a kimenetei már a kisebb-egyenlő-nagyobb relációkat is jelzik, és több ilyen áramkör összekapcsolható 4 bitnél hosszabb számok összehasonlítására. Ebben az esetben az előző fokozat kimeneteit kapcsoljuk a következő fokozat bemenetéhez (ún. kaszkádba kapcsolás).

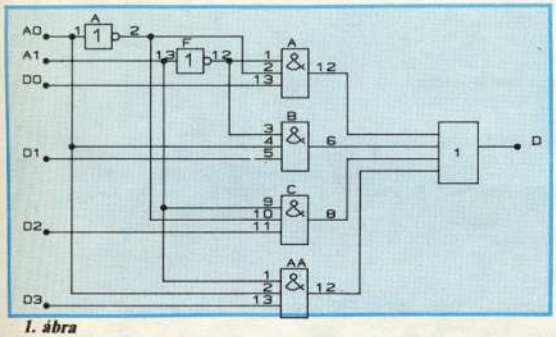
Számláló áramkörök

A jelek (impulzusok) számlálása tulajdonképpen a tárolás és az összeadás (kivonás) műveleteire bontható. A számláló áramkörnek valamilyen kódban tárolnia kell a már megszámlált impulzusok számát, majd az újabb impulzus érkezésekor ehhez 1-et hozzáadni (előreszámlálás) vagy 1-et kivonni (hátraszámlálás) kell a tárolt értékből. A számlálók egymással összekapcsolt flip-flopokból épülnek fel. A számlálók kivezetései funkcionálisan négy csoportba sorolhatók:

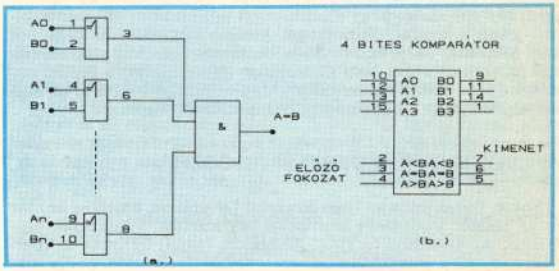
— tápfeszültség-kivezetések,
— a számlálandó jel (órajel),
— a számláló állapotát tartalmazó kimenetek,
— egyéb, a működést meghatározó jel: a

ejlesztési eszközök

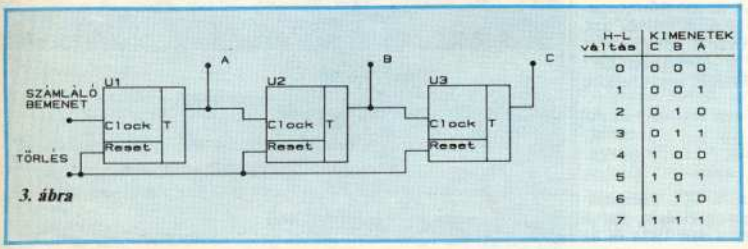
A sorozat alapgondolata — azon a régi felismerésen kívül, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza. A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozónak nélkülözhetetlenek az alapfokú áramköri hardverismeretek is. Megerősíti ezt, hogy szaporodik az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.



1. ábra

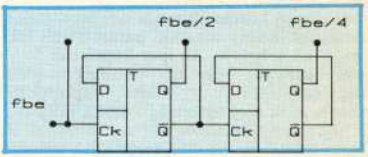


2. ábra

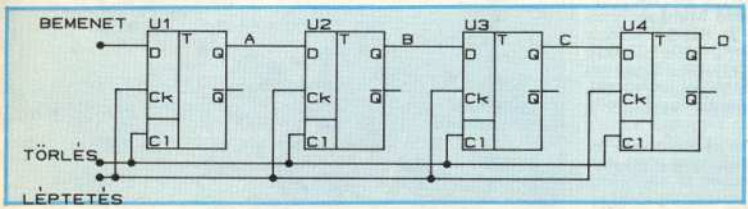


H-L váltás	KIMENETEK C	B	A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

3. ábra



4. ábra



5. ábra

számlálót alapállapotba hozó törlő jel (RESET), a számlálás irányát meghatározó jel stb.

A működési mód alapján megkülönböztetünk szinkron és aszinkron számlálókat. A számlálás az órajel felfutó (pozitív élvezérlés) vagy lefutó (negatív élvezérlés) élének hatására következik be.

A leggyakrabban használtak a kimenetükön a számlált értéket bináris, illetve BCD kódban megjelenítő számlálók. Számlálókat mind TTL, mind CMOS technológiával gyártanak. Az elérhető nagyobb elemsűrűség miatt több tárolóelemet tartalmazó („hosszabb") számlálókat lehet CMOS technológiával kialakítani. Például a TTL család SN7490-es típusjelű tagja 4 bites BCD számláló, a CMOS család CD4040 típusú tagja 12 bites bináris számláló.

Példaként a *3. ábrán* egy háromfokozatú bináris előreszámlálót mutatunk be. Az ilyen típusú számlálóknál minden egyes flip-flop kimenet a következő flip-flop órajel bemenetére csatlakozik. Az esetek többségében a számlálás előtt a számlálóláncot nullázni kell, hogy a számlálás nulláról induljon. Ehhez a számlálót (illetve a tárolókat) ki kell egészíteni egy RESET vagy CLEAR bemenettel, amivel a számlálót (a tárolókat) alaphelyzetbe állíthatjuk.

Tételezzük fel, hogy az R jelű törlő (RESET) bemenetekre adott törlő impulzussal megtörtént az A, B, C bemenetek alaphelyzetbe állítása. Amikor a számláló bemenetére érkezik az első H-L átmenet, az U1 tároló kimenete H szintet vesz fel (billen). A kapcsolás most számlálta az első jelet, így állapota ØØ1. Az U1 bemenetén az impulzus végét jelző L-H átmenet hatására semmi sem történik a negatív élvezérelt működés miatt. Az újabb H-L átmenet U1-et visszabillenti alaphelyzetbe, de akkor a kimenetén — ami U2 bemenete — a H-L átmenet U2 flip-flopot billenti, és így tovább. Az ábrán a kimenetek állapotai a táblázatból jól nyomon követhetők.

Osztók

A számlálók előnyös tulajdonsága, hogy kimenetükön a bemenő jel leosztott frekvenciával jelenik meg. Például az előző ábrán A, B, C kimeneteken egyre feleződő jel jelenik meg. C-nél négy L, négy H szint, B-nél két H, két L, A-nál egy H, egy L szint. Ilyen módon, számlálókból kialakított láncokkal egy adott frekvenciájú bemenő jelet leoszthatunk. A *4. ábrán* két D flip-flopból kialakított frekvenciafelező, illetve negyedelő kapcsolás látható.

Léptetőregiszterek

Szintén flip-flopokból épülnek fel a tárolt digitális információ jobbra, illetve balra léptetését (shiftelését) megvalósító léptetőregiszterek. Amikor egy léptetőimpulzus beérkezik, akkor az információ az egyik tárolóból a következő tárolóba lép. Egy ilyen léptetőregiszter működése az *5. ábrán* követhető végig.

Tegyük fel, hogy az U1 D tároló bemenetére H szintet adunk. Ilyenkor semmi sem történik mindaddig, amíg az órajel (léptetőimpulzus) nem érkezik meg. Amint az órajel megjelenik, az U1 flip-flop bemenetén levő H szintű jelet egy flip-floppal jobbra, az U2 flip-flop bemenetére lépteti, és ez így folytatódik az újabb impulzusok hatására.

(Folytatjuk)

DR. KÓNYA LÁSZLÓ

„A számítástechnika mindenkié, a számítástechnika mindenkiért" szlogen jegyében harmadízben rendezték meg az Országos Mikroszámítógépes Találkozót. Dömöki Bálint, az NJSZT elnöke megnyitó beszédében hangsúlyozta, hogy a Tavaszi Fesztivál keretében megtartott találkozó elsősorban céljában különbözik az ország többi számítástechnikai kiállításától. A rendezők olyan kiállítókat kértek fel, akik a nagyközönség számára használható termékeket mutattak be, és olyan programokat szerveztek, amelyek széles érdeklődésre tarthattak számot. Ezért is alakult a kiállítás összetétele úgy, hogy a hazánkban először bemutatott VIDEOTEX rendszer és NOVELL hálózat mellett igen nagy számban megtalálhattuk az egyéni és kisvállalkozókat is.

III. Országos

Humánus bit

Igazán egyéni színfoltot jelentett a „Gép is ember" karikatúrakiállítás. A zsűri díszelnöke, Heinz Zemanek professzor ünnepi beszédében elmondta, hogy elgépiesedett világunkban „humánus bitet" jelent a humor. Mosolyogni komoly dolgokon is lehet, sőt kell. Mint már 88/5. számunkban beszámoltunk róla, a kézzel rajzolt pályázatok közül Jurij Koszobukin (Szovjetunió), a géppel rajzoltak közül pedig Halász Géza (Magyarország) nyerte el az első helyet. A fődíjat Vjekoslav Klaic (Jugoszlávia) kapta.

Amiről a profik csak álmodoznak...

Sokan voltak jelen az Ipari-Szakmai Délutánon, amelyen az ipar és az amatőr mozgalom képviselői beszélgettek, vitatkoztak egymással. Az amatőrök (a HCC mozgalom tagjai) felrótták az iparnak, hogy nem érdeklődnek műszaki fejlesztéseik iránt, pedig sokszor megelőzik a vállalatokat. A HCC-nek például már akkor saját fejlesztésű mikroszámítógépe volt, amikor az ipar csak álmodott róla. A közelmúltban is kifejlesztettek egy kötegelt adatátviteli rendszert, amely műszaki paramétereit tekintve a világon bárhol megállná a helyét. Az ipar képviselői felhívták az amatőrmozgalom figyelmét, hogy egy készülék megalkotása nem jelenti azt, hogy sorozatban is gyártható. Megkérdezték továbbá, hogy miért nem keresik meg újdonságaikkal a vállalatokat és szövetkezeteket? A vita azzal zárult, hogy az amatőröknek sem idejük, sem érzékük nincs üzletelni, ellenben felajánlják tudásukat.

Ipari programok pályázata

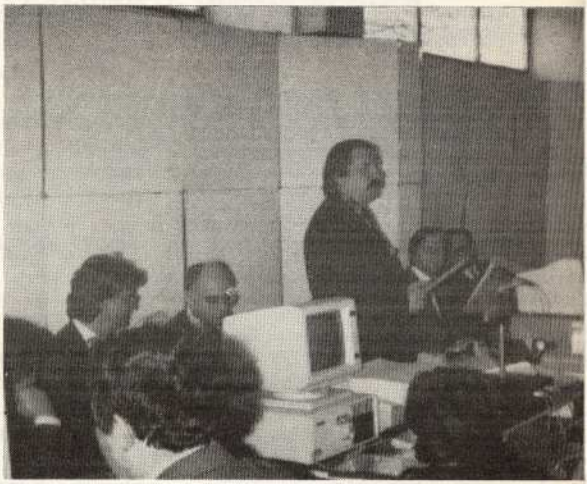
Az érdeklődők megnézhették az IPM és az NJSZT Számítástechnikai Klubja által szervezett szoftverpályázat zsűrizését, illetve bemutatóját. Az ipar területén használt és kifejlesztett IBM és az azzal kompatibilis gépeken futó programok versenyeztek. Első lett és ezzel tizenötezer forintot kapott a Csepel Művek Szerszámgépgyárában készített NC/CNC Alkatrész-előkészítő Munkahely elnevezésű program. A bemutatót Karai Csaba és Sándor Károly tartotta. A fejlesztők valóban az életből merített kérdést oldották meg. A bírálóbizottság két második díjat adott át. Nádor Vilmosné és Dunai Péter, a Csepel Autógyár dolgozója a Technológiai Darabjegyzék Kisszámítógépes Adatkezelő Rendszere, valamint a DÉMÁSZ-tól Szilágyiné Jancsó Ilona és Benke Gábor Telefőkönyv, Főkönyvi Könyvelés programja vitte el a hétezerötszáz forinttal járó második díjat. (A díjakat az Ipari Minisztérium ajánlotta fel.)

Váradi László, a pályázat szervezője elmondta, hogy a bemutatott hat szoftver magas színvonalú volt. Különösen szerencsés, hogy a gyakorlati életből merített problémák megoldására készültek CAD/CAM, illetve adatfeldolgozásra. Meglepő azonban a pályázók alacsony száma.

Házi feladat telefonhívásra

A széles nyilvánosság első ízben láthatta működés közben az Ipari Minisztérium IPCON és az osztrák MUPID VIDEOTEX rendszerét.

A magyar rendszert a MUPID és a Siemens segítségével alakították ki. Jelenleg a hálózatot ötven vállalat használja, de előreláthatólag háromszáz vállalat kapcsolódik később hozzá. A központi gépet a VEIKI üzemelteti, a vállalatoknál elhelyezett terminálok



Ipari-szakmai délután

Heinz Zemanek professzor megnyitja a karikatúrakiállítást

Mikroszámítógépes Találkozó

telefonvonallal csatlakoznak hozzá. Nagyon fontos, hogy a vonalakat csak híváskor foglalja le. Ezzel a rendszerrel az amúgy is gyengén ellátott vállalatok nem vesztenek vonalat.

Csak jelszó ismeretében lehet a rendszer funkcióit használni és információt kérni. Természetesen nem mindenki jogosult minden szolgáltatás használatára. A bejelentkezés után az alapmenüből választhatók ki a különböző funkciók. Ezek közül csak néhányat említünk meg. Alapszolgáltatásként a tulajdonosok statisztikai adatokat, időjárásjelentést, valutaárfolyamokat stb. kérdezhetnek. Ezek köre természetesen az adatállománytól függ. A szervezők is meglepődtek, hogy a vállalatok milyen szívesen használják az üzenetküldés funkciót, aminek segítségével üzenhetnek meghatározott állo-

A nyertesek

másnak, és a rendszerbe bekötött cégeknek is. Praktikusnak tartom, hogy a jövőben a vállalatoknak nem szükséges levélben elküldeniük jelentéseiket, elég ha azt beviszik az IPCON-ba. Sőt, a minisztérium is szétküldheti tájékoztató adatait a VIDEOTEX-en. Nem szükséges kiadványokat szerkeszteni! A vállalatok betáplálhatják elfekvő készleteiket és az bármelyik terminálon lekérdezhető. Elfoglalt vezetők is szívesen küldenek üzenetet más cégeknek az időrabló telefonálgatás helyett. A terminál önmagában is intelligens, programozható. Így igen jól használható a teleprogram funkció, amelynek segítségével számítógépes programok hívhatók be és a terminálon futtathatók. A Magyar Posta országosan bevezeti a VIDEOTEX rendszert, ami jelenleg huszonhat országban működik.

Ezt követően megismerkedhettünk az osztrák VIDEOTEX rendszerrel, a MUPID-dal is. A MUPID-ról lapunk előző számaiban már közöltünk ismertetést (87/10-es számunktól a 88/2-ig tartó sorozatban), de felejthetetlen élmény volt működés közben látni. Hermann Mauer Grazból, a rendszer egyik alkotója mutatta be a különböző szolgáltatásokat. Láthattuk a telemenetrendet, a telefonkönyvet, a telexküldést, telelexikont, telejogszabály-gyűjteményt és az elektronikus újságot. A MUPID szolgáltatásai teljességre és emberközpontúságra törekednek. A telefonkönyvnek például van egy asszociációs része, ha nem emlékszünk pontosan a keresett névre, akkor megadja a hasonló hangzásúakat és írásúakat.

A MUPID szorosan kapcsolódik az osztrák távoktatáshoz is. Hívásra a „távhallgató" megkapja a soron következő oktatóprogramot, amellyel gyakorolhat. A házi feladatot, a leckét is a rendszer küldi, amit a tanuló visszaküld a központnak értékelésre.

Bridzs és sakkszimultán a számítógéppel

A találkozón a szellemi sport is jelen volt. A bridzsrajongók új játékprogramokkal vívhattak. Újdonságok voltak az IBM kompatibilis gépekre készült bridzsprogramok.

A sakk-kedvelők örömére Bilek István nagymester szimultánt adott a Saitek sakkszámítógépnek. Tizenegy táblán játszottak. A nagymester nyolc győzelmet és két döntetlent ért el. Egy alkalommal a gép bizonyult jobbnak, vagy Bilek nagymester játszott gyengébben? A játszmák kiértékelése további szórakozást nyújtott e játék szerelmeseinek.

Óvodásoknak sem korai

A 24-es pavilon előadóterme egyszerre gyermekzsivajtól lett hangos. Ekkor kezdődött ugyanis a LOGO bemutató. Seymour Papert professzor, a LOGO programozási nyelv kifejlesztője sajnos igérete ellenére sem tudott eljönni, de Heinz Zemanek átvállalta a házigazda szerepét. A zsűri díszelnöke szakértelmével és közvetlenségével mindenki szívébe belopta magát. Óvodás korú gyerekek ültek a számítógépek előtt és mélyültek el a LOGO titkaiban. Amint Zemanek professzor elmondta, a gyerekek minden félelem nélkül közelítenek a számítógéphez. Nincs bennük olyan ellenérzés, mint a felnőttekben, hogy „nem értek hozzá". A gyerekek kipróbálják és megjegyzik a történéseket. Olyan dolgokra jönnek rá ezáltal, amelyekre a felnőttek képtelenek. Játszva tanulni, ez a képekről a legegyszerűbb.

Egészségügyi informatika

A találkozón tíz egészségügyi intézmény, valamint az egészségüggyel együttműködő EISZI mutatta be termékeit. A kiállítást a szervező vállalat egymáshoz kapcsolódó folyamatok szerint tervezte meg. A beszélő számítógépen kívül megismerkedhettünk a komputertomográf és ultrahangvizsgáló szimulációs készülék működésével. A diagnosztikai oktató programcsomag nagy hasznára van az orvostanhallgatóknak. A program kérdés-felelet formájában gyakoroltatja a diagnózis felállítását.

Itt mutatták be a nem beszélő, súlyos mozgássérült gyerekek vizsgálatára kialakított képességmegállapító és fejlesztő programot.

A találkozón bemutatott közegészségügyi, gyógyszerészeti nyilvántartó rendszerek bevonultak a gyógyászatba. Segítik az oktatást, tehermentesítik az orvosokat és gyógyszerészeket, pontosabbá teszik a nyilvántartásokat.

Sütés-főzés számítógéppel

A szervezők a pályázat meghirdetésekor nem hittek igazán a verseny sikerében. Meglepődtek, amikor — igaz kisszámú — változatos és magas színvonalú pályázatok érkeztek. Hatan mutatták be produkciójukat. Első díjat és ezzel háromezer forintot Burján János kapott a „MENÜ Étlapszerkesztő" programjáért.

A „Menü"-vel három speciális szempont alapján kiválasztott étlapot lehet összeállítani. (Például felnőttek, sportolók és gyerekek részére.) A program figyelme szinte mindenre kiterjed. A beépített öröknaptár figyeli a hétvégéket, a gasztronómiai ellenőr meg az összeférhetetlen ételeket utasítja vissza. Az adatbankban ételféleségek szerint, mint leves vagy zöldség, megtalálhatók a nyersanyagok, azok árai és összetételük. Így a főszakács ügyelhet a menük árára, de kalória, fehérje és szénhidráttartalmára is. Jelenleg huszonötféléből, száz-száz ételt tart nyilván. Ez azonban szinte korlátlanul bővíthető, mivel a program teljes adatbázis-kezelést valósít meg.

A zsűri két második díjat adott ki. Két-kétezer forinttal jutalmazták Borka Zoltán „Receptek" és Nagy Imre „Kevés anyagból változatos ételek" elnevezésű programját. A „Receptek" ebben a témában újszerűségével tűnt ki. A képernyőn nemcsak az étel elkészítéséhez szükséges nyersanyagok, hanem az elkészítéséhez szükséges műveletek ábrái is megjelennek. Nagy Imre programja először megkérdezi, hogy mit akarunk, fogyni vagy hízni, és milyen alapanyagunk van otthon? Ezek alapján összeállítja az elkészíthető ételeket.

Örömmel láttuk, hogy az egysíkúnak hitt témát milyen változatosan oldották meg a versenyzők. Igaz, itt is akadt rossz példa, ami sajnos az oktatóprogramok versenyén is előfordult. Egyik versenyző nem tett mást, mint egy szakácskönyvet hatalmas munkával egy az egyben feldolgozott számítógépre. Ez sajnos nem a gép tudásának a legjobb kihasználása.

Morzézik a számítógép

A Háziépítésű Eszközök Országos Versenyére tizennégy pályázat érkezett, s ebből tizenketten mutatták be alkotásukat. Kiemelkedő volt és első helyezést ért el a Petrik Lajos Vegyipari Szakközépiskola On-Line mérési rendszere. A hatezer forintos díjat a KFKI ajánlotta fel. A diákkör kémiai mérésekhez fejlesztett ki egy speciális interfészt. A méréseket értékelő szoftver is nagy teljesítmény, azzal is díjat nyerhettek volna. A sokoldalú mérési rendszerből most csak a térfogat-megállapítást emeljük ki. A folyadék térfogatát úgy állapítják meg, hogy azt csepegtetik és egy fotokapu-érzékelőn keresztül a cseppeket a számítógép számolja, közben figyeli a folyadék PH értékét. Az IPM két különdíját kapta a Kisújszállási MHSZ Rádióklubja és Ecsedi Csaba.

A kisújszállásiak amatőr rádiókészülék vezérlésére dolgoztak ki programot és interfészt. A program menürendszeréből kiválasztható például az általános hívás, gyors billentyűzet vagy a gépbe írt szöveg adása. A morzézés sebessége is megadható. Teljesítményük azért is nagyszerű, mert csak tavaly ősz óta van a klubnak számítógépe.

Ecsedi Csaba sorstársainak, a csökkentlátóknak készített interfészt a telefon számítógépes vezérlésére. A gép figyeli a hívást, bejelentkezik és elindítja a telefonhoz kapcsolt egyszerű magnetofont. A gép telefonálni is tud, a Homelab bemondja a beprogramozott szöveget. A készülékkel a gyengénlátók kapcsolatot tarthatnak egymással.

Nem bírták a kisgépek

Az Oktatóprogramok Országos Versenyére több mint kétszáz pályázat érkezett. A mikroszámítógépes találkozón százötven pályamunkát mutattak be. A tantárgyak a következők voltak: fizika, humán- és szakmai tárgyak, matematika, nyelv, biológia, kémia és egyéb. Idén is a szakmai tárgyakban születtek a legjobb oktatóprogramok, míg más tárgyakban helyben topogás vagy visszaesés tapasztalható. Újdonság volt a menüválaszték és a felhasznált adatállomány növekedése. Ez azzal is magyarázható, hogy a legtöbb pályázat hajlékonylemezen érkezett, a kazetta lassan kiszorul. A versenyen bemutatott programokról később számolunk be, most csak a csapatverseny eredményét ismertetjük. Első helyezést kapott és a KSH által felajánlott húszezer forintos díjat a Landler Jenő Szakközépiskola. Második lett és a SZÜV BASIC oktatócsomagját nyerte a Móricz Zsigmond Gimnázium csapata. A zsűri két harmadik díjat ítélt oda. A SZÁMALK könyvcsomagját kapta a Szentesi Művelődési Ház és a Kilián György Gimnázium csapata. (A csapatok eredményét az egyéni teljesítmények alapján állították össze.)

Az oktatóprogramok bemutatóján először tapasztalhattuk, hogy a kisgépek mennyire nem alkalmasak huzamosabb igénybevételre. Az idő haladtával egyre több gép romlott el. A zsűrinek ennyi hardverproblémája a megelőző években nem volt. Ahogy a gépkocsipark öregszik országszerte, úgy szaporodnak a gondok a kisszámítógépekkel és a perifériákkal is. A számítógépek nemcsak erkölcsileg, hanem fizikailag is avulnak.

A szervezők szándékosan március 21-re, iskolaszüneti napra tették az ifjúsági napot, hogy minél több tanuló vehessen rajta részt. A HUNGEXPO azonban riasztó ajándékkal lepte meg őket: ezt a napot szakmai nappá nyilvánította. Így ötvenforintos belépőjegyet kellett váltaniuk a tanulóknak és nem fogadták el a diákigazolványokat sem. Ezzel kizárták az ifjúságot! Nekünk természetesen a Mikroszámítógépes Találkozó ifjúsági rendezvényeiről való távolmaradásuk fájt a legjobban, de az ÁISH sportpavilon és a játékkiállítás szervezői sem örültek ennek.

PINKE GYÖRGY

Szimultán a sakkszámítógéppel

Azért az ember is gondolkodik

Az INTELL-GRAF szövegszerkesztő rendszer

Múlt havi számunkban bemutattuk Olvasóinknak a TELE-SCRIPT telefonáló-szövegszerkesztő rendszert. Akkor főleg a telefonáló részegységek felépítésének ismertetésére fektettük a hangsúlyt. Most bemutatjuk a TELE-SCRIPT rendszer köré épített, általános szövegszerkesztő rendszer főbb sajátosságait.

Az INTELL-GRAF szövegszerkesztő az EASY SCRIPT rendszer teljes magyar adaptációja: tartalmazza az összes magyar ékezetes betűt. A Commodore 64 számítógép billentyűzetét a szabvány magyar betűelhelyezés szerint használhatjuk segítségével (tehát aki például vakon gépel, annak a keze magától a megfelelő helyekre talál), és biztosítja az eredeti EASY SCRIPT összes szokásos funkcióját.

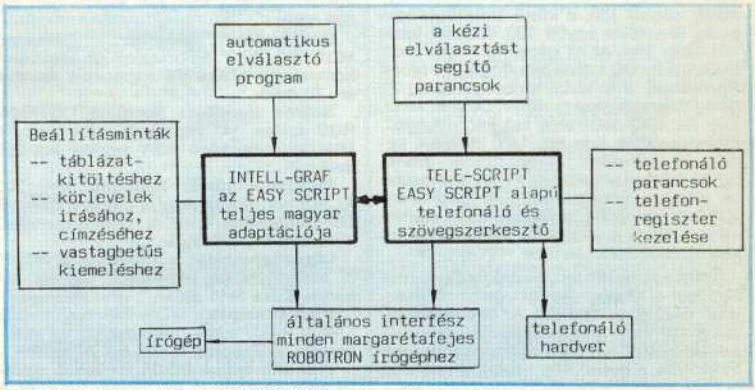
Ezzel eddig nem sokat mondtunk a rendszerről: az EASY SCRIPT-nek létezik legalább tízféle magyar adaptációja. A számítástechnikai lapok olvasói kikupálódhattak a különböző szövegszerkesztő rendszerek összehasonlításában, számos teszt jelent meg. Az érdemi összehasonlítás azonban hovatovább illuzórikussá vált, különösen ha csupa EASY SCRIPT alapú rendszert hasonlítunk össze. A tesztelők által adott jegyek eltéréseit csak apró eltérések okozzák. Többszörös mélytengeri búvárvizsgával rendelkező specialisták gondos aprómunkával próbálják lemérni, hogy melyik is a súlyosabb hiányosság: az, ha az egyik rendszerben az aláhúzás bekapcsolása kicsit nehézkesebb, vagy az, ha a másikban a pontosvesszőt nehezebb előcsalogatni.

A gyakorlat azt mutatja, hogy ezek az apró különbségek nem jelentenek lényeges előnyt vagy hátrányt. A rendszerek mindennapos, üzemszerű használatában az apróbb, könnyen kikerülhető hiányosságokat a felhasználó tökéletesen megszokja, mint kutya a bolháját. Ösztönszerűen elkerüli azokat a kellemetlenségeket, amelyek a tesztelőknek jogos bosszúságot okoznak. Elárulunk például, hogy még az angol eredeti (ékezetek és írógépvezérlés nélküli) EASY SCRIPT is teljesen le tud fagyni, úgy, hogy a képernyő teljesen kimerevedik és a gép csak kikapcsolás után indítható újra.

Nem rosszindulatú (tesztelői) búvárkodás közben találtunk erre a hibára, hanem úgy, hogy amíg nem ismertük eléggé a rendszert, nem volt kialakult munkamódszerünk, ergonómiánk a használatához, addig mai szemmel nézve elképesztő logikátlanságokat csináltunk. Amikor a program nem pontosan úgy viselkedett, ahogy a saját fejünk alapján elképzeltük, ijedtünkben összevissza csapkodtuk a billentyűket, például egy sorban több sorvégjelet is létrehoztunk, és ez okozta a rendszer lefagyását.

Valamennyi általunk ismert magyar EASY SCRIPT alapú rendszerben azóta is benne van ez a hiba, és egyetlen felhasználót sem zavar komolyan: a rendszer tudatos, célszerű használata közben még véletlenül sem kerül elő. Pedig az a tény, hogy a rendszer rendeltetésével nem ellenkező használat közben is tökéletesen ki tud akadni, sokkal súlyosabb hiba, mint a tesztelők által elemzett nüanszok bármelyike. Mégis joggal rendkívül népszerűek ezek a rendszerek, mert ha valaki egyszer megtanulta használni őket, nagyon kényelmessé teszik a mindennapi irodai munkát. Nincsen komplex program hiba nélkül, de az nem mindegy, hogy a normál mindennapi használat során lépten-nyomon beleütközik-e a felhasználó a hibákba, vagy a hibák csak különlegesen logikátlan használat esetén kerülnek elő.

Az elválasztás problémájának megoldását annyira fontosnak találták a rendszer tervezői, hogy az INTELL-GRAF mellett például a „felette" szót: ehhez érteni kellene a teljes mondat jelentését, hogy most arról van-e szó, hogy valaminél magasabb helyen, vagy minden táplálék elfogyasztásáról. A megértés problémája a mesterséges intelligencia jelenlegi állása szerint alapjában nincs megoldva. Mindazonáltal (hasonlóan a mesterséges intelligencia számos egyéb eredményéhez, például a szakértői rendszerekhez), a megoldás ebben az esetben az igényeket bőségesen kielégíti.



Az INTELL-GRAF és a TELE-SCRIPT rendszer fő elemei

Az INTELLROBOT által kifejlesztett INTELL-GRAF szövegszerkesztő rendszer is beilleszkedik az EASY SCRIPT magyar adaptációinak sorába, az alapvető funkciókban az eltérések kicsik, és általában az „ízlés dolga" kategóriába tartoznak. Maga az alapprogram tehát csak kevéssé különbözik a többitől. Az egész rendszerben viszont jó néhány olyan új vonás található, amelyek lényegesen kellemesebbé teszik a mindennapi munkát.

Az egyik fő ilyenfajta kiegészítés a magyar helyesírás szabályainak megfelelő automatikus elválasztást biztosító program. Aki huzamosabban használt már szövegszerkesztőt, az tapasztalhatta, hogy milyen szemet strapáló, nehézkes munka az elválasztgatás. Ezen segít ez a program. A feladat bizonyos értelemben a mesterséges intelligencia területéhez tartozik, és eredménye, mint csepp a tengert, mutatja a mesterséges intelligencia mai állása szerint elérhető eredményeket.

Az elválasztás az esetek jó 99 százalékában tökéletes. Jól választja el például a program az asszony szót, de (ugyanerre a mintára) hibázik a vasszeg szónál. Ezt a fajta hibát még jórészt ki lehetne kerülni egy nagyszótár beprogramozásával, jóllehet az eredmény aligha érné meg a program lassulását. Százszázalékos eredmény azonban még így sem lenne elérhető. Válasszuk el működő TELE-SCRIPT rendszerbe is beépítettek néhány olyan új parancsot, amelyek a kézi elválasztást nagymértékben kényelmesebbé teszik. A TELE-SCRIPT rendszer egy egyszerűsített EASY SCRIPT alapú szövegszerkesztő, amely fájl szinten kompatibilis az INTELL-GRAF rendszerrel. A TELE-SCRIPT fő tulajdonságait múlt havi számunkban mutattuk be, lényege, hogy a szövegszerkesztés közben a rendszer automatikusan képes telefonhívásokat is végezni. Az egyszerűbb szövegszerkesztési feladatok teljes mértékben elvégezhetők a TELE-SCRIPT segítségével is, bonyolultabb formátumparancsokat igénylő szöveg pedig telefonálás közben is elkészíthető így, majd a végső formátumot az INTELL-GRAF segítségével alakíthatjuk ki. A két rendszer egymástól függetlenül is használható, de igen szerencsésen egészítik ki egymást.

Egy másik típusú kiegészítő szolgáltatás az INTELL-GRAF rendszerhez, hogy a szövegszerkesztő bonyolultabb funkcióit kész beállításmintákkal támogatja. Ezzel nemcsak könnyebbé teszi a táblázatok kitöltéséhez, körlevelek írásához és címzéséhez, szép íráskép kialakításához használható bonyolult funkciók megértését, hanem azonnal hozzáférhetővé is teszi a számítógépes észjárásban járatlan felhasználó számára.

—i.r.—

Áruk és áruk

1987 októberében tartotta amerikai partnerlapunk, a Rainbow a 14. fesztiválját. Ezen a rendezvényen évente kétszer a Tandy cég által gyártott Color Computer (CoCo) sorozat gépeit, valamint az azokhoz készített hardvert és szoftvert állítják ki, vitatják meg. A 14. fesztivál — melyen több mint tízezren vettek részt — szenzációját a Tandy üzletlánca, a hétezer boltból álló Radio Shack hálózat árai jelentették.

A Commodore 64-gyel egy időben kidolgozott CoCo2 ára 30 dollár (kb. 1500 Ft), ugyanez 16 k RAM esetén mindössze 10 dollár, a tavaly megjelent, a C128-nak megfelelő CoCo3 ára 100 dollár volt.

Kétoldalas lemezegységet is árultak 45 dolláros áron. A 0,5 Mbájtos bővítést egy másik cégnél 100, a külső lemezegységet pedig illesztővel együtt 130 dollárért lehetett kapni. Egy, az XT gépeknél megszokott EGA felbontású színes monitoros, 0,5 Mbájt RAM táros, 360 kbájt lemezegységes, 20 Mbájt merevlemezegységes CoCo3 rendszer ára 1240 dollár (egy hasonló kiépítésű, XT-kompatibilis gép kb. 1450 dollárba kerül). A rendszerre írt szoftverek száma ugyan messze elmarad az XT/AT-kompatibilis gépekre készültekétől, de Unix-szerű (OS—9) operációs rendszere miatt már inkább a nálunk professzionális gépeknek nevezett kategóriába tartozik.

Két évvel ezelőtt kezdte meg a Computer Shopper c. szaklap „Az év legjobb bevásárlása" nevű díjak kiadását. Az 564 színes oldalon (!), több mint negyedmillió példányban megjelenő havilap a díjak odaítélésének indoklását, a gyártó cég nevét és az árakat is közli. Mi az alábbiakban mindebből csak a termékeket soroljuk fel, árukkal együtt.

Számítógéprendszer: Northgate A—Turbo. AT-kompatibilis rendszer, 8-10 MHz órafrekvenciával, 1 Mbájtos RAM tárral, 360 k és 1,2 M kapacitású lemezegységgel, 65 M (!) kapacitású merevlemezegységgel, egy soros és két párhuzamos csatolóval, 12" képméretű, ámbraszínű monokrom monitorral. Ára 2000 dollár, ami magában foglal egy olyan egyéves garanciát, hogy az elromlott alkatrészt 24 órán belül cserélik.

Olcsó számítógép: Amiga 500. Ára 589 dollár.

Számítógép: Ultra-Comp Turbo. A 256 k RAM tárral, AT-szerű dobozban, 5150 típusú billentyűzettel, 360 k kapacitású lemezegységgel, monokrom monitorral, monokrom/grafikus kártyával ellátott gép ára 539 dollár.

Monokrom monitor: Samsung MD1252G. A 12", ámbraszínű, ún. Hercules-kompatibilis (729 × 350 képpontú) monitor ára 79 dollár.

Színes monitor: Samsung CD1464W RGB színes, 14" képméretű, dönthető, 14 MHz sávszélességű EGA monitor ára 259 dollár.

Mátrixnyomtató: Seikosha SP—1200 AI. A 9 tűs, 120 karakter/s (levélminőségnél 25) sebességű 2 × 3 k pufferkapacitású nyomtató ára 159 dollár.

Lézernyomtató: a „Merre tart a világ?" c. sorozatunkban ismertetett Okidata Laserline 6 ára 1449 dollár.

Bővítőkártyák: MCT—MF hatfunkciós (384 k RAM, soros, párhuzamos és játékcsatoló, óra/naptár) kártya ára 79 dollár.

Videokártya: a BOCA Research videokártyája egyaránt jó monokrom CGA, EGA monitorokhoz és fényceruza-bemenete is van. Ára 199 dollár.

Lemezegység: nincs megjelölve az ára.

Merevlemezegység: Miniscribe 3000 Plus. A 20 M kapacitású, 53 ms hozzáférési idejű, automatikus fejbeállású egység ára 319 dollár.

Kiárusítás: a Xerox 16/8 kétprocesszorú (8086 és Z80), 128 k RAM kapacitású, 360 k-s lemezegységes, 10 M kapacitású merevlemezegységes, két soros és egy párhuzamos csatolójú, 12" monokrom monitoros rendszer ára 349 dollár. Ebben a CP/M, a CP/M—86, az MSDOS, a WordStar, a SpellStar, a Star Index és a MailMerge szoftver is benne van.

Könyvelési szoftver: az ACT—1 teljes rendszer ára dokumentációval együtt 100 dollár. Sajnos a könyvelési rendszerek különbözősége miatt itthon nem használható.

Integrált szoftver: az Integrated 7 Package a hasonló című sorozatunkban közölteken felül grafikus és terminált helyettesítő programot is tartalmaz. Ára 68 dollár.

Táblázatkezelő: a Twin egyenértékű a LOTUS 1—2—3 Release 1A változatával. Ára 38 dollár.

Szövegszerkesztő: a Leading Edge szövegszerkesztő 30 dollárért kapható.

Irodai nyomtatószoftver (Desktop Publishing): Pagemaker ára 545 dollár.

Adatbázis-kezelő: a közismert dBase III Plus ára 389 dollár.

Nyelv: a MIX—C Works csomagot, amely tartalmaz egy Editor, egy C compiler, egy Ctrace és egy ASM utility programot, 39 dollárért árulják.

Grafika: a Dr Halo II ára 60 dollár. SE

KLUBLAPOKBÓL

Biztos benne (I/N)?

Egyszer valaki le akart törölni a lemezéről egy állományt, amelyre már nem volt szüksége. Az állomány, hogy védve legyen a felülírástól, a szövegszerkesztő egy alállományában volt elhelyezve. Emberünk az állományt FRANCIS.DOC-nak nevezte, így annak teljes megnevezése C:(WP)FRANCIS.DOC volt. Ennek megfelelően beírta:

DEL C:(WP) FRANCIS.DOC

Nem vette észre, hogy végzetes hiba keletkezett a parancsban: a parancsbeadás közben a FRANCIS.DOC elé egy szóköz csúszott be. Bárcsak tudta volna, hogy amikor a DOS megkérdezi:

ARE YOU SURE (Y/N)?

akkor valójában azt is közli, hogy: ön globális törlésre készül. Mivel emberünk biztos volt abban, hogy törölni akarja a FRANCIS.DOC-ot, Y-t válaszolt. A DOS alaposan tanulmányozta a parancsot a hibáig, és készségesen megsemmisítette a WP állományban levő összes fájlt.

Most azt gondolhatjuk, hogy piszkos dolgot művelt a DOS, pedig valójában ez kétélű kérdés. Néha nagyon kényelmes ezt az eljárást használni. A

CD/TEMP után
DEL *. * majd
CD/ végül
RD TEMP helyett beírhatjuk, hogy
DEL TEMP majd
RD/TEMP.

De jusson eszünkbe, hogy a DOS a teljes törlés jóváhagyását kéri, amikor ezt látja:

ARE YOU SURE (Y/N)?

(Írta: Martin Shulman. Megjelent a Valley Computer Newsletter című lapban. Fordította: Lucza János.)

µINFORM

Rovatunkban az Apple, Atari, Commodore és Sinclair mikrók tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről informáljuk olvasóinkat egy tartalomleíró szólánc segítségével.

A forráshely karaktersorozatát nyíl vezeti be, ezt a / jelig a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalszámát szerepelteljük. A második / jel után pedig - az esetleges másolatkérést megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A tartalomleíró szóláncok permutálásával és alfabetikus rendezésével szerkesztett teljes anyagot az OMIKK "APACS" c. kiadványsorozata tartalmazza (vevőszolgálati telefon: 341-765). Lapunk ebből csak a "programlista" címszóval kezdődő részletet teszi közzé.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek az OMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. Másolatok a SZÁMALK-tól csekkszelvény, az OMIKK-tól megrendelőlevél beküldésével, vagy személyesen rendelhetők.

	A folyóirat neve	Kódja
x	64'er Magazin	64er
	Antics	anti
x	Chip Magazin	chip
x	Compute!	cute
x	Dr. Dobb's Journal	dobb
	Elektor Electronics	etor
	Happy Computer	happ
x	mc - Zeitschrift	mc
	Run (USA)	run
	Run (NSZK)	run2
x	Your Computer	your
x	ZX Computing Monthly	ZXCM

PROGRAMLISTA
ql spectrum adatcsere szubrutinok
->zxcm/87.03-66/5
PROGRAMLISTA
ql spectrum zene v i billentyuzet mint hangszer klaviatura (tune maker)
->zxcm/87.01-12/5
PROGRAMLISTA
ram-bank barkacsolas bovites c64 eprom operacios rendszer atkapcsolas alaparamkor beegetes alkalmazasi utmutato ->run2/87.02-16/5
PROGRAMLISTA
ram-floppy barkacsolas cp/m mikroprocesszor (68000) latteres 8-rol 16 bites aramkor az ec-buszhoz
->mc/87.01-59/10
PROGRAMLISTA
ram-floppy bovites c64 vezerloprogram 2x64k-hoz ->run2/87.01-86/4
PROGRAMLISTA
ram-floppy c64 utasitaskeszlet bovites (businessbasic)
->run2/87.03-114/10
PROGRAMLISTA
spectrum barkacsolas merestechnika kapacitasmeres a/d atalakitas nelkul
->happ/87.03-166/2
PROGRAMLISTA
spectrum basic programsorok atszamozasa (renum) ->happ/87.03-186/1
PROGRAMLISTA
spectrum cad grafika 3-d vetuleti rajzok ->zxcm/87.02-78/3
PROGRAMLISTA
spectrum cad grafika egyenes/kor/ellipszis rajzolas ->zxcm/87.01-86/5
PROGRAMLISTA
spectrum compiler hisoft-pascal (autostart) ->happ/87.02-111/2
PROGRAMLISTA
spectrum feltores hibakereso segedletek ->zxcm/87.02-20/3
PROGRAMLISTA
spectrum fileazonositok kiiratasa (mc header reader) ->your/87.02-65/1
PROGRAMLISTA
spectrum filekezeles lemezegyseg (microdrive) helykimelo tarolas gyorstoltes ->zxcm/87.02-72/3
PROGRAMLISTA
spectrum gepi kod programok hibamentesitese (debugger)
->happ/87.03-187/1
PROGRAMLISTA
spectrum grafika jatekprogram keszitesterkep-hatter hasznalata
->zxcm/87.01-32/5
PROGRAMLISTA
spectrum grafika kepernyokezeles gorgetes abra/karakter pozicionalas
->your/87.02-62/3
PROGRAMLISTA
spectrum grafika kepernyokezeles nyomtatas pontok/savok kezelese
->zxcm/87.03-22/4
PROGRAMLISTA
spectrum jatek (guess the pair)
->your/87.03-64/2
PROGRAMLISTA
spectrum jatek (kingdom of kull)
->zxcm/87.02-60/6
PROGRAMLISTA
spectrum jatek (ppowerhouse)
->your/87.03-61/4
PROGRAMLISTA
spectrum jatek (the war of the shires) ->zxcm/87.01-75/4
PROGRAMLISTA
spectrum jatekprogram keszites regies karaktertipusok kiemelt iniciale (illuminator) ->zxcm/87.01-18/4
PROGRAMLISTA
spectrum kepernyokezeles ablakhasznalat 1.resz ->zxcm/87.01-66/4
PROGRAMLISTA
spectrum kepernyokezeles ablakhasznalat 2.resz ->zxcm/87.02-75/3
PROGRAMLISTA
spectrum kepernyokezeles gorgeto rutinok ->zxcm/87.02-66/5
PROGRAMLISTA
spectrum kepernyokezeles tarcim-tablazat hasznalata ->zxcm/87.03-58/2
PROGRAMLISTA
spectrum lemezegyseg (discovery) kezelo rutinok ->zxcm/87.01-70/4
PROGRAMLISTA
spectrum lemezegyseg (discovery) kezelo-rutinok ->zxcm/87.02-24/4
PROGRAMLISTA
spectrum lemezegyseg (discovery) uzemeltetesi tanacsok ->zxcm/87.03-84/3
PROGRAMLISTA
spectrum segedrutin gyujtemeny
->zxcm/87.01-51/3
PROGRAMLISTA
spectrum sprite grafika jatekprogram keszites demo ->zxcm/87.01-42/3
PROGRAMLISTA
spectrum szovegfeldolgozas (specword) 1.resz ->zxcm/87.01-28/3
PROGRAMLISTA
spectrum szovegfeldolgozas (specword) 2.resz ->zxcm/87.02-42/4
PROGRAMLISTA
spectrum szovegfeldolgozas (specword) 3.resz ->zxcm/87.03-50/4
PROGRAMLISTA
spectrum szovegkezeles jatek kiegeszites az (illuminator) hoz demo
->zxcm/87.02-12/5
PROGRAMLISTA
spectrum tarszervezes grafika levedes tarolas szekvencialis filekent
->zxcm/87.01-49/2
PROGRAMLISTA
spectrum zene (spectrum 128 sound)
->your/87.03-72/3
PROGRAMLISTA
spectrum zene hangminta-tarolas/modositas (sound sampler)
->zxcm/87.03-32/5
PROGRAMLISTA
spectrum zene midi illesztes
->zxcm/87.01-37/1
PROGRAMLISTA
speedscript atari xl/xe segedprogramok ->cute/87.02-49/2
PROGRAMLISTA
sprite apple ii megjelenito-utkozteto utasitasok definiciotablazat tarolasa ->mc/87.02-59/2
PROGRAMLISTA
sprite c16 grafika pszeudo-spriteok shape-pel editor demo
->run2/87.02-120/3
PROGRAMLISTA
sprite c16 plus/4 pszeudo-spriteok (shape) modusokkal
->run2/87.03-106/2
PROGRAMLISTA
sprite c64 grafika hypra-basic hires utasitasok funkcionalis modulok
->64er/87.01-83/5
PROGRAMLISTA
sprite c64 jatekprogram keszites basic bovites (sprite-access)
->run2/87.03-132/5
PROGRAMLISTA
sprite c64 jatekprogram keszites demo ->run2/87.01-92/3
PROGRAMLISTA
superbase c64 alkalmazasi utmutato 6.resz modul gyujtemeny
->64er/87.03-171/7
PROGRAMLISTA
szamrendszer atari xl/xe dec-bin-hex atalakitas ->cute/87.02-73/3
PROGRAMLISTA
szamrendszer c64 dec-bin-hex-ascii atalakitas (zahlenwandler)
->happ/87.02-61/1
PROGRAMLISTA
szovegfeldolgozas atari st beszedkimenet kiegeszites a (talking typewriter)-hez ->anti/87.01-75/4
PROGRAMLISTA
szovegfeldolgozas atari xl/xe fileformatum olvashatosag elemzese (the san francisco fogger)
->anti/87.02-28/4
PROGRAMLISTA
szovegfeldolgozas c128 nyomtato makrok a (runscript 128)-hoz 2.resz
->run/87.01-78/2
PROGRAMLISTA
szovegkezeles c64 teljes szo atvitele sorvegekent basicbol hivhato szubrutin ->run/87.01-84/3
PROGRAMLISTA
tarszervezes c64 gepi kod ml-programok kezdo es vegcimenek kikeresese
->cute/87.03-68/1
PROGRAMLISTA
tarszervezes c64 grafika hires kepek karakterkent valo kezelese
->happ/87.03-46/2

Jobb verem Pascalban

Az Olvastunk rovatban a Magazin 1987/9. számában láthattunk egy Pascalban megvalósított verem szerkezetű memóriát. A cikk tetszett, de a program nem. Mit kifogásolok? Először is teljesen felesleges két logikai változó. Mivel minden veremhivatkozásnál mindenképpen meg kell vizsgálni a verem állapotát, úgy vélem, nyugodtan elhagyhatjuk. A másik (talán még súlyosabb) stiláris hiba az, hogy a műveletek globális változókon keresztül kommunikálnak. Azért vannak a paraméterek, hogy használjuk őket!

Ezek után egy lényegesen jobb veremmegvalósítást javasolok (lásd a *listát*). Ez blokkokból áll. Ha az aktuális blokk betelt, újat nyitunk, így a veremnek csak a halom maximális mérete és a többi dinamikus változó szab határt. Jól felismerhető a láncolt struktúra. A láncolás kétirányú, ami alapjában véve felesleges, de így a létrehozott és később kiürített veremterületet nem kell újra elkérni a rendszertől. A memóriafoglalás drága művelet, igaz, ez kis gépeken általában nem számít, de a nagyokon fontos lehet.

A javasolt megoldás négy eljárásból áll. A NEW_BL a veremlistához illeszt új blokkot. Erre az eljárásra az inicializálásnak és a PUSH műveletnek van szüksége, de mi is hívhatjuk, például ha a futás elején fel akarunk állítani egy n blokkból álló vermet. A STACK_INIT inicializálja a vermet. Ezt az eljárást akár a főprogram elejére is tehetnénk önálló utasításokként, de így jobban áttekinthető.

A PUSH művelet a paraméterként kapott értéket helyezi a verembe. Ha nem sikerül végrehajtani a műveletet, akkor betelt a halom (ez a verem fizikailag a halomra — HEAP — kerül), ebben az esetben vagy a fordító megfelelő opcióját kell megváltoztatni, ha van, vagy a program más megoldását kell felkutatni. A művelet sikerességének ellenőrzése gépfüggő, ezért hagytam el.

A POP műveletet függvényként valósítottam meg. Ha üres veremről próbálunk levenni elemet, a POP FALSE értékkel tér vissza, egyébként TRUE-val, és paraméterében visszaadja a verem felső elemét. Ha a POP blokkhatárt lép át, a kiürült blokkot nem törli, hogy az esetleges következő PUSH hívásnál ne kelljen újra lefoglalni a következő blokkot. Természetesen ha nagyon kell takarékoskodni a memóriával, a törlést is beépíthetjük.

A főprogram a rutinok tesztelését végzi. Feltölti a verem egy blokkját, és még egy számot tesz a veremre, majd ezeket kiolvassa és kiírja. Az utolsó hívásnál hibaüzenetet kapunk.

Azt hiszem, ennél lényegesen jobb általános célú veremmegvalósítás nem sok létezik. Sebességét kétségtelenül csökkenti a mutatókezelés, de cserébe a veremnek csupán a hardver szab határt.

TÖRÖK TIBOR

```
program STACK_DEMO( Output );

const
  BL_SIZE = 100;

type
  BLOCK = packed array
            [ 1..BL_SIZE ] of INTEGER;
  STACK_P = ^STACK_BL;
  STACK_BL = record
               Bl : BLOCK;
               Bl_p : INTEGER;
               Next, Prev : STACK_P
             end;

var
  Stack : STACK_P;
  I, J : INTEGER;

procedure new_bl( var Last : STACK_P );

var
  Slave : STACK_P ;

  begin
  new( Slave );
  Slave^.Prev := Last;
  Slave^.Next := NIL;
  if Last = NIL then
    Last := Slave
  else
    Last^.Next := Slave;
  Slave^.Bl_p := 0
  end;

procedure stack_init;

  begin
  Stack := NIL;
  new_bl( Stack )
  end;

procedure push( I : INTEGER );

  begin
  if Stack^.Bl_p = BL_SIZE then
    begin
    if Stack^.Next = NIL then
      new_bl( Stack );
    Stack := Stack^.Next
    end;
  with Stack^ do
    begin
    Bl_p := Bl_p + 1;
    Bl[ Bl_p ] := I
    end
  end;

function pop( var I : INTEGER )
                        : BOOLEAN;
  begin
  if ( Stack^.Prev = NIL ) and
       ( Stack^.Bl_p = 0 ) then
    pop := FALSE
  else
    begin
    pop := TRUE;
    with Stack^ do
      begin
      I := Bl[ Bl_p ];
      Bl_p := Bl_p - 1
      end;
    if Stack^.Bl_p = 0 then
      if Stack^.Prev <> NIL then
        Stack := Stack^.Prev
    end
  end;

  begin
  stack_init;
  for I := 1 to BL_SIZE + 1 do
    push( I );
  for I := 1 to BL_SIZE + 2 do
    if pop( J ) then
      writeln( J )
    else
      writeln('* * * Stack hiba. * * *')
  end.
```

OLVASTUNK...

könyvet, novellát, olvasói véleményt és úgy döntöttünk, hogy ebben a cikkünkben szólunk néhány szót magáról e rovatról. N. Wirth „Programming in MODULA—2" (Springer Verlag, 1983.) könyve alapján bemutatjuk a veremtár egy lehetséges megvalósítását.

E rovatról

E rovatot éppen két éve indítottuk. Akkor még nem tudtuk, meddig (szeretjük?) csináljuk majd. Nem írtunk bevezetőt sem: úgy gondoltuk, a sorozat beszéljen önmagáért. Most — ezzel néhány olvasói észrevételre is referálva — röviden pótoljuk ezt a hiányt.

Kiindulásunk az volt, hogy Magyarország a számítástechnikában (is) a világ *perifériáján* helyezkedik el: tipikusan *nem* nálunk történnek a lényeges, a világ (a technika) fejlődését meghatározó dolgok. Ezért jó, ha vigyázó szemeinket — lehetőségeinkhez mérten — a centrum(ok) fejlődésben kulcsszerepet játszó műhelyein tartjuk: egy-egy jellemző gondolat felvillantásával felhívjuk olvasóink figyelmét egy-egy kiemelkedően jónak talált műre, igyekezve elkerülni a szakmainak álcázott, de lényegében kereskedelmi propagandacélokat szolgáló forrásokat. Tisztában vagyunk azzal, hogy olvasóink többsége — már csak pénztárcájának lapossága miatt is — eléggé nehezen fog hozzájutni a bemutatott könyvekhez. Így számára az egy-egy cikkben elférő kerek gondolat gyakran többet ér, mint egy-egy tartalmi összefoglaló vagy átfogó minősítés.

```
DEFINITION MODULE Buffer;
 EXPORT QUALIFIED put, get, nonempty, nonfull;

 VAR nonempty, nonfull: BOOLEAN;
 PROCEDURE put(x: CARDINAL);
 PROCEDURE get(VAR x: CARDINAL);
END Buffer.

IMPLEMENTATION MODULE Buffer;
.....
PROCEDURE put(x: CARDINAL);
BEGIN
 IF n < N THEN
  buf [n] : =x;n:=n+1;nonfull:=n < N; nonempty:= TRUE
 END
END put;

PROCEDURE get(VAR x: CARDINAL);
BEGIN
 IF n > 0  THEN
    n:= n-1;x:=buf [n]; nonempty:= n>0;nonfull:= TRUE
 END
END get;
BEGIN n:= 0;nonempty:= FALSE; nonfull:= TRUE
END Buffer.
```

Bár közlünk jócskán programozási példákat is, ezek célja elsősorban és főleg nem az, hogy *receptet* adjunk: *elveket* szeretnénk e példák segítségével megvilágítani. (A piacon számos olyan könyv van forgalomban, mely *kipróbált* és jó hatásfokú algoritmusokat tartalmaz. Ezek közül az egyik legismertebb és legjobb Knuth „Fundamental Algorithms" című könyve: The Art of Computer Programming, Volume 1, Addison—Wesley Second Editions, 1975.). Tudjuk, valamely elv bemutatására alkalmasnak látszó program általában még *nem eladható termék*, de a Bernoulli-törvény (hatás) szemléltetésének sem a legcélszerűbb eszköze a kereskedelmi forgalomban kapható modern porlasztó, és viszont: senkinek sem jutna eszébe kocsijába egy demonstrációs eszközt beépíteni. (Ez nem zárja ki annak lehetőségét, hogy egy műszakilag jó megoldás egyaránt alkalmas demonstrációs célokra — támaszul egy elv jobb megértéséhez — és egyúttal „kereskedelmi forgalomképes" is.)

Gondolataink leírására — a nemzetközi gyakorlattal összhangban — mi is (forrásaink is) gyakran használtuk a Pascal nyelvet. Ezt világszerte megteszik azok is, akik általában nem Pascal-programozással keresik mindennapi kenyerüket. A „profi paszkálosok" nem szoktak megharagudni ezért: különbséget tesznek a csupán gondolatközlésre és az „éles", kereskedelmi felhasználásra szánt programok, programrészletek között.

A példákat — bár ez a szöveg kényes volta (pontosvesszők, kettőspontok stb!) miatt mind a szerkesztőségnek, mind a nyomdának nagy nehézségeket, e rovat írójának pedig többször nem kis boszszúságot okozott már — a jobb olvashatóság érdekében általában nem listákról fotózott ábrákon közöltük, hanem kiszedettük (egészen a legutóbbi időkig). A kezdeti tapasztalatok alapján a szerkesztőség szigorított a „technológiáján", de — sajnos — még mindig elég sok az értelemzavaró hiba. A szerző optimista és a maga részéről kitartana a szedés mellett. Ő javulást tapasztal.

Ismét a veremtárról

A Magazin 1987/9. számában J. Haugelandnak a MIT Pressnél 1986-ban megjelent könyve kapcsán Neumann János alapvető felismeréséről írtunk. Bemutattuk, mekkora jelentősége volt a programok és az adatok egyöntetű tárolásának a szubrutinokból való visszatérésnél. Említettük, hogy az általános megoldást, a *veremtárat* csak később, az 50-es évek közepén találta meg A. Newell és C. Shaw. Mivel nem voltunk biztosak abban, minden olvasó tudja-e, mi az a veremtár (ismerik-e az olvasók magát az elvet), közöltünk egy Pascalban írt példát a verem megvalósítására, mely sajnos (itt nemcsak a nyomda és a szerkesztőség hibájából) hibásan jelent meg. A hibákért elnézést kérünk. Az elvet az alábbiakban N. Wirth nyomán most már MODULA—2-ben mutatjuk be.

—KE—

Ren nagyon feldúltan érkezett vissza a városba. Nem nézett semerre, még a köszönéseket sem fogadta. Berontott a házba és végigvetette magát az újonnan készített ágyon. Harag és félelem kavargott benne. Mindez abból fakadt, hogy nem értette a Gép viselkedését.

— Nem értem! — morfondírozott magában. — A Gép tulajdonképpen parancsokat osztogat nekem! Nekem, egy élőlénynek. Ha nekem parancsol, akkor rajtam keresztül az összes polgárnak parancsol egyúttal! Hát legyen. Egy gép nem lehet okosabb, mint én! Igen, azt mesélte, isten náluk testet öltött. Hát legyen isten. Imádni fogjuk és ajándékokkal halmozzuk el. De nem árthatja bele magát az életünkbe.

Fejébe húzta a cilinderét és elindult a Géphez. Útközben többször átgondolta és felépítette mondanivalóját. Elérve a kockát, megtorpant. Még soha nem érezte ilyen taszítónak a Gépet.

— Mit akarsz? — harsant fel a Gép, és Renben egy pillanat alatt összekuszálódott minden, a beszéd, amit kimunkált, a gondolatai, melyeket oly nagy gonddal rendezgetett.

— Eljöttem hozzád, hogy bocsánatot kérjek. Szükségünk van rád! Rádöbbentünk a nagyságodra, amelyhez képest mi csak apró porszemek vagyunk, és maradhatunk mindörökre.

Ren egyre jobban belejött. Már-már úgy érezte, visszatér az előre elhatározott kerékvágásba. De a Gép közbevágott.

— Nektek még szükségetek van rám. Egyszer majd belátod magad is. Közös immár a célunk.

— Ugye nem akarsz semmi rosszat? — kérdezte Ren.

— Nem akarok semmi rosszat.

— Tajnak igaza volt? — képedt el Ren. — Te vezetni akarsz bennünket?

— Csak amíg el nem érjük a célt.

— Egy gép? Gépként akarsz uralkodni felettünk, élőlények felett? Értelmes élőlényeket akarsz leigázni? Azért, mert van fegyvered? A Földön sem sikerült, pedig biztos vagyok, hogy ott is arra törekedtél. Lehet, hogy miattad robbant fel az a bolygó! Ott is csak eszköz voltál az emberek kezében. Nem tudom elképzelni, hogy a Földön Gépek irányították volna az élőlényeket!

— Sajnos nem így volt. Talán akkor elkerülhető lett volna a katasztrófa. Ilyen sehol sem alakulhat ki. Itt sem.

— Hát hogy lesz? — üvöltött Ren. — Hogyan? Ellentmondásba keveredtél saját magaddal, azt hiszem, hogy egy ilyen gépnél, mint amilyen te vagy, ez a legnagyobb hiba. Nem?

— Te is gép vagy Ren.

A vezető ennek hallatára megdermedt. Hirtelen újult erővel támadt fel benne a düh.

— Igen, a világmindenség pedig egy hatalmas DNS-lánc, ami végtelenül kanyarog. És . . . — szavait félbeszakította a jelenés. A hengertestű, félgömbfejű, három csápot lebegtető egykori „ellenség" megjelent a szerelőfolyosó felett. Renre nézett és gondosan artikulálva a hangokat, összeállította a szót: — Bionom! Bionom.

— Mi az, hogy bionom? — nyögte amikor magához tért.

A Gép nem felelt. Az ajtó egy szisszenéssel feltárult és a kocka belsejét betöltötték az éjszaka neszei. Ren nem mozdult. A Gép vetíteni kezdte a víz alól szerzett első tekercset. Ren némán gubbasztott a fal mellett, a Gép sem törte meg a csendet. A film lepergett. A Gép vetíteni kezdte a háborút megörökítő képeket.

— Mégis — szólalt meg Ren halkan —, hogyan éltük túl a háborút?

— A fegyverek mindig túlélik a háborút, ha meg is fogyatkoznak közben. Figyelj csak! — mondta a Gép, amikor feltűnt a bionomokat bemutató rész. — Látod, őket neveztétek ellenségnek. Pedig csak annyi bűnük volt, hogy megépítettek benneteket. Ott az „ellenség" mögött a képen, látod? Talán a nagyapád! Vagyis lehet, hogy van benned belőle egy alkatrész. — A Gép a legédesebb hangszínét használta. — Csak eszközök voltatok az élőlények közötti polgárháborúban. Csak eszközök!

— Miért vagyunk képesek tanulni? — kérdezte Ren, utolsó erejével. Az ellenkezés tüze már régen kialudt benne.

— Valószínűleg az alkotóitok hihetetlenül jól bővíthető memóriákkal láttak el benneteket. Az algoritmus, ami a fogalmak összekapcsolását hajtja végre, nagyon nagy adattömeg kezelésére képes. Csak nem volt elég adatotok. Csak azt tudtátok, hogyan kell bányászni, karbantartani önmagatokat, és ilyeneket. A háborúban rengeteg új információt szedtetek fel. Lehet, hogy nélkülem is eljuttattok volna a cél megvalósításáig. De velem gyorsabb lesz! Kiküszöbölöm mindazt, ami konfliktust idézhetne elő.

— A cél! Mi a cél? Mi az amiért . . . — elakadt a bionom. Képtelen volt befejezni a mondatot.

— A célotok az én célom is! Együtt várunk majd egy értelmes lényt, hogy hirdessük neki, élt az ember!

— Kicsoda? — kérdezte Ren.

— Az ember.

— Ezt a valamit nem ismerem! — Ren befejezettnek tekintette a beszélgetést. Megfordult, és a város felé irányította lépteit.

Ruhái, melyekre nem is olyan rég annyira büszke volt, már széttépve borították a padlót. Ren megállt és a tükörben vizsgálta magát. Fényérzékeny szemei kitágultak. Egyenként felemelte mind a hat csápját. Összedörzsölte a vájatok mélyítésére szolgáló karmait. Mindegyik csápja végén kettő-kettő, ilyen egymással szemben álló karom villogott hófehéren. Leemelte a falra

erősített tükröt. Hosszúkás kocsonyás hasáb bámult vissza rá. Csápjával egy helyen benyomta a fejét, a mélyedés hosszú percekig ott maradt, és csak nagyon lassal töltötték ki ismét a testnedvei. Sokáig vizsgálta magát. Hirtelen a karmok összezáródtak a tükör széleinél, és az üveg száz darabra szétpattanva hullott a talajra.

— Nem, a polgárok ezt nem tudhatják meg! — ebben biztos volt.

— Elpusztítani! — vágott bele. De ez nem lehet megoldás. — Hiszen annyi mindent tanulhatnánk még tőle. Akkor magammal végzek! De az csak rontana a többiek helyzetén. Én már úgy ahogy tisztában vagyok a helyzettel, de velük elölről kezdi majd. Hogyan bizonyosodhatnék meg arról, hogy hazudik? Hogyan bizonyíthatnám be magamnak, hogy nem vagyok Gép? Az állag, az anyag nem döntő. Marad az értelem. Az intelligencia. A mesterséges és a született. Mi a különbség? Mi? A következtetés módja. Statikus, ez az! Az ismeretek bővülhetnek, de ahogy egymás mellé sorakoztatom, az ha ugyanolyan, akkor gép vagyok, egy mesterséges intelligencia. De ezt nem tudom eldönteni. Ezt magáról senki nem tudja eldönteni!

Letelepedett a padlóra. Csápjait maga köré tekerte. Agyából a gondolatok utolsó foszlányai is elillantak.

Reggel a polgárok motozása riasztotta fel Rent. Kinézett az épület előtti térre, és csodálkozva látta, hogy valamennyi társa ott áll és várja.

— Elmegyek és beszélek a Géppel! — mondta, de hangja idegenül csengett még önmaga számára is. — Utána mindent megbeszélünk! Amikor a bolygónk napja pont felettünk áll majd, gyertek utánam!

A tömeg néma maradt. Ren megfordult és fénylő csíkot húzva maga után, újra a kocka felé indult. Bízott benne, hogy utoljára. Lassan haladt. Gondosan megfontolva minden lépést. Agyába véste az út minden egyenetlenségét. Kétszer annyi idő alatt ért fel a kockát a város elől eltakaró dombra, mint máskor.

Az első esetet leszámítva, most volt először zárva a bejárat, ami a Géphez vezetett. Ren komótosan a zárt ajtóhoz sétált. Csápját az ajtóra helyezte, hófehér karmaival megkaparta a bejáratot.

— Beszélnünk kell! — kiabálta.

A Gép ellenőrizte, hogy egyedül van-e, és miután meggyőződött erről, felnyitotta a kaput. Ren belépett.

— Miért jöttél? — kérdezte a Gép.

— Beszélnünk kell egymással.

— Hallgatlak.

— Egész éjjel azokon a dolgokon gondolkodtam, amiket mondtál — kezdte Ren. — Azon gondolkodtam, hogy miképp bizonyosodhatnék meg arról, hogy valóban gép vagyok. De rádöbbentem, hogy felesleges. Akár élőlények vagyunk, akár mesterséges szerkezetek, te mindenképpen uralkodni fogsz rajtunk. A többieknek nem szóltam, mégis csak jobb élőlényeken uralkodni! Nem?

— De igen.

— Bár neked, vagy annak aki alkotott, azt hiszem teljesen mindegy, hogy gépek, vagy élőlények! Minket is élőlénynek néztél. És nem volt igazad. Vagy igen? Most már ennek sincs semmi, de semmi jelentősége. Rájöttem, az embereknél jobb alanyok leszünk. Mi képtelenek lennénk elpusztítani önmagunkat. Egy gép erre nem képes!

— Igaz, csak ha erre építik meg — vetette közbe a Gép.

— Mi már régen nem léteznénk, ha az önpusztításra lennénk alkotva. Ez belátható. Az embereknél jobbak leszünk . . . — Ren nem fejezte be a mondatot. Előrevetette magát és a vájatmélyítésre tervezett karmok belevájtak az egyik lézertároló védőlemezébe. A bionom egyetlen rántással letépte és a falhoz vágta.

A Gép Ren mozdulatának pillanatában elindította a szerelőrobotot. A fémtest nyikorogva közelítette meg a tomboló lényt. Ren megfordult és minden erejével a közeledő robotra sújtott. Ami megtorpant egy pillanatra, de — igaz horpadt előlappal — folytatta a bionom megközelítését.

Ren megfordult, és amilyen gyorsan csak tudott, menekülni kezdett a szerelőfolyosók kacskaringós szövevényén. A belső érzékelők kullancsként rátapadva követték és minden mozdulatát közvetítették a Gépnek. A Gép aktivizálta a másik robotot is és megpróbálta Rent beszorítani egy sarokba. Ren majdnem a kocka tetejéig nyúló tárolók között haladt. A csikorgás megszűnt. Ő is megállt, és hallgatózott. Egy keresztúthoz érve, hirtelen elhatározással balra fordult. Az útról ahonnan kilépett, felharsant a fémes csikorgó zaj. Tudta, hogy a visszafelé vezető utat is elvágták. Egyetlen irány maradt. Arra vetette magát, de szinte a levegőben megdermedt. Az ellenség várta, csápjait maga körül lebegtetve. Túl közel volt, hogy menekülőösztöne parancsolhatott volna. Karmaival megpróbálta megragadni az előtte lebegő lényt, de a semmit markolta. A kép abban a pillanatban eltűnt.

Már rájött arra, hogy a számítógép központi egységéhez kellene férkőznie, de a robotok pontosan az ellenkező irányba szorították. Vaktában csapkodni kezdett. Karmai áthatoltak a védőborításokon. Úgy látszik az egyik ütése hatásos volt, mert fehér szikrák pattantak elő a Gép testéből. De csak a belső érzékelők egyik kapcsolódobozát tette tönkre. Erejét megfeszítve felkapaszkodott egy mátrixszekrényre. És nem mozdult.

A kockában teljessé vált a csend egy pillanatra, hogy szinte egyből utána felhangozzon a robotok csikorgása. Ren a hangokból tudta, hogy két üldözője külön utat választott magának.

A szerelőfolyosón csakhamar feltűnt az egyik robot. Ren megvárta, amíg pontosan aláér. Két csápját beleakasztotta a szekrénybe, biztosítva ezzel a visszavonulását, majd ráugrott a fémszerkezetre.

Egyszerre három pár karom markolt bele, és szakította át a robotot védő burkot. Ren, negyedik csápjával igyekezett távol tartani magától a fémtüskét. Sokkal erősebb volt, mint a robot. Felhúzta a mátrixszekrény tetejére. A második robot ekkor fordult be a folyosóra az ellenkező irányból.

Ren a közeledő géphez vágta a csápjai között szorított robotot.

Úgy látszott, a két gép egymást semmisítette meg. Összegabalyodott testüket elektromos kisülések hagyták el. Majd kesernyés füstöt árasztva mozdulatlanul maradtak a földön.

Ren lemászott a perifériáról és a Gép villogó részéhez ment.

A Gép vak és süket volt már, kivéve egy kis részt, ami a mérnöki panel előtti területre korlátozódott.

Ren megállt a villogó fények előtt. A Gép szédületes tempóban fogott hozzá a helyzetelemzéshez.

Ren csápja felemelkedett, és belevágott a központi egységet védő fémlemezbe.

A Gép kiadta a parancsot. A műhold ráállt a megadott koordinátákra és tüzelt. Az energianyaláb, az egyetlen erő ami képes volt áthatolni a kocka falán, tűhegynyivé szűkülve, becsapott a kockába, és szinte széthajtogatta annak oldalait.

A polgárok összerezzentek a robbanásra. A Gép felé néztek, és hitetlenkedve bámulták az ég felé törő fekete füstoszlopot.

Epilógus

. . . igyunk arra, hogy mi emberek nem tűnünk el nyomtalanul a semmibe. Hagyunk magunk után valamit, ami megmutatja a világmindenség bármely értelmes lényének, hogy milyen volt az ember! Talán ez a Gép egy kicsit olyan mint mi, hiszen mi alkottuk. Úgy gondolom, remek munkát végeztünk! Igyunk hát a Gépre! Cheerio! — és George Keden, akit a Gép Mesterként tisztelt, szájához emelte a poharát.

Vége

Király plusz gyalog a király ellen III.

Az első két részben eljutottunk odáig, hogy a „jósló" algoritmust megvalósító programot bemutathassuk. Mivel a legtöbb olyan gyalogelszabadulási pozíció, amely több lépést igényel, nem ismerhető fel egyszerű szabállyal, más megoldást kellett találnunk. Arra hagyatkozunk, hogy figyelembe vehetjük: ezek olyan állásokhoz vezethetnek, amelyeket programunk tartalmaz. A pattnak és a gyalog kiütésének értékét definícióban határoztuk meg: ezek döntetlent jelentenek.

Hogyan harcoljunk?

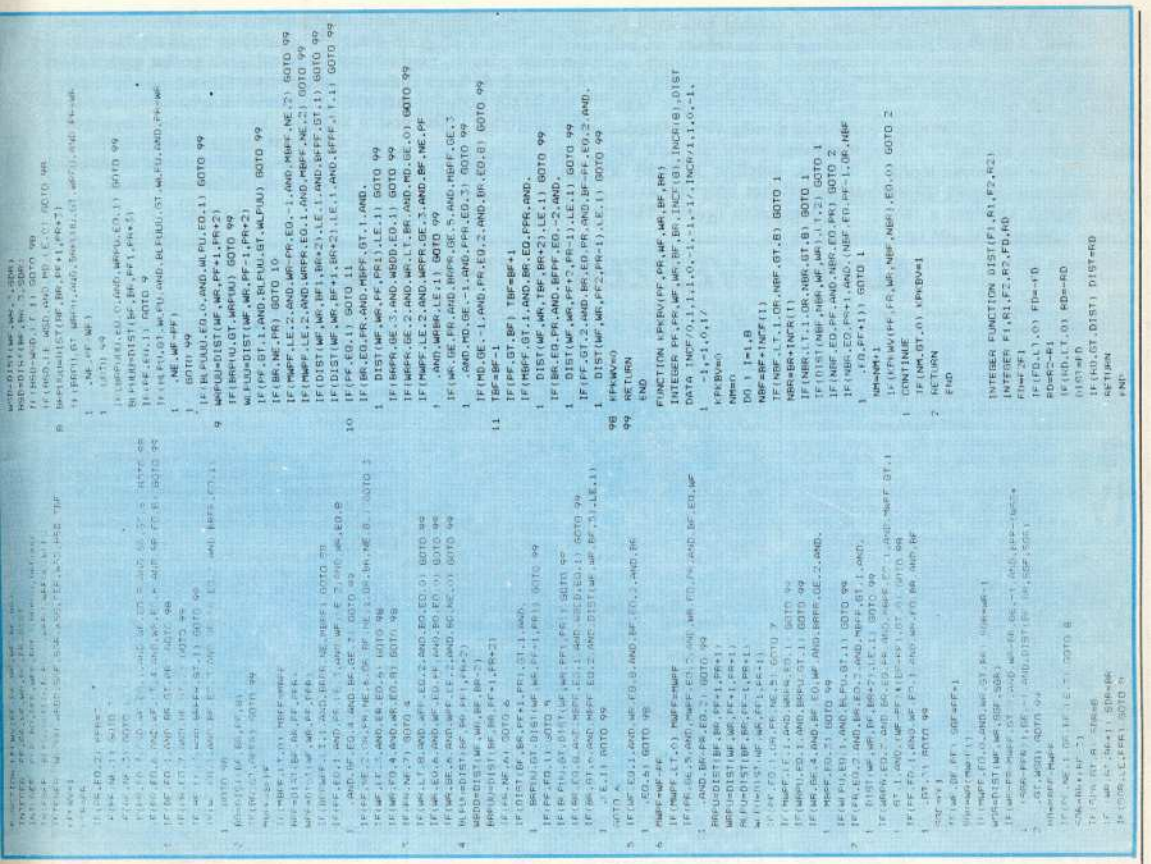
Az algoritmus a játékfa felépítésénél összehasonlítja az illető pozíciót a már ismert állásokkal, és ha talál az ismertek között a jelenlegivel azonos hadállást, akkor azt a fa végpontjaként kezeli és kiértékeli. A királytávolság figyelése sokat segít abban, hogy könnyen felismerjük a már ismert pozíciókat. Nagyon sok hadállásról meg tudjuk állapítani, hogy ismertek-e, illetve azt, hogy nem ismertek, ha a király nagyon messze van azoktól a fontos mezőktől, kulcsmezőktől, amelyek már egy ismert állást reprezentálnak. Tehát ezeket az eseteket ki lehet zárni.

Az *1. táblázat* a logikai szabályok feltételeinek reprezentációja, amellyel megkapjuk az eredményt, hogy nyerő-e a pozíció vagy sem. Ez lesz a következőkben az alapfüggvényünk, amelyet KGKVE-vel jelölünk a Király+Gyalog a Király ellen Világos Értékelése szerint kifejezés kezdőbetűiből. Ez egy feltételtáblázatot és logikai kifejezéseket tartalmaz. Negyvennyolc szabály van, amit a táblázatban a számok alatt láthatunk.

Ha a program bármelyik szabályt alkalmazhatónak találta, akkor az algoritmus a megfelelő W vagy D (angol terminológia szerint: nyerő

1. táblázat

| | | |



vagy döntetlen) értéket rendeli a függvényhez és azonnal kilép a szubrutinból. Ha nincs alkalmazható szabály, akkor az eredmény döntetlen. A szabály akkor és csak akkor alkalmazható, ha az összes feltétel teljesül. Vagyis az egy oszlopban levő feltételek a logikai ÉS műveletthez vannak összekapcsolva, amíg a különböző oszlopok oszloponként összetartozó feltételeire a VAGY művelet igaz.

A táblázat felső része egyszerű formában adja meg a különböző feltételeket, míg az alsó részben bonyolult logikai kifejezések alkotják a többi feltételt. A *2. táblázatban* láthatjuk, hogy egy tesztsorozat folyamán melyik szabályt hányszor kellett alkalmazni.

Az 1. táblázat jelölései az angol terminológia szerint a következők:

P	a gyalog mezője
W	világos király
B	sötét király
F	utótagként használva: vonal
R	utótagként használva: sor
PP	=P kivéve, ha PR=2. Ez utóbbi esetben, amikor a gyalog alaphelyzetben áll a 2. soron, akkor a gyalog előtti mezőt jelenti
Q	az átváltozási mező, például QF=PF, QR=8
→	az egyes mezők közötti távolság például: W→P=max (/WF-PF/, /WR−PR/)
L	bal oldali mező, például LP=a gyalog melletti bal oldali mező
R	(előtagként) jobb oldali mező, lásd: bal oldali mező
+	az adott mező előtt levő mező, például [P+]=a gyalog előtt levő mező
−	a mögötte levő mező
[ ]	mező,] például [RP+]=átlósan jobbra egy mezővel
[f, r]	sorral és vonallal határozzuk meg a mezőt
T	egy vonal a gyalog fele, például TBF=if BF>PF then BF−1 else BF+1
SD	az alábbiak szerint meghatározott mező: SDF=3; SDR=if WR<=BR+1 then BR+BF−3 else BR
SG	az alábbiak szerint meghatározott mező: SGF=if WF<PF then PF-1 else PF+1 SGB=if WF=PF&WR>BR then WR−1 else WR−/WF−PF/+1

A táblázat a mezők meghatározására mind a sorszámnál, mind az oszlopszámnál 1-től 8-ig terjedő koordinátákat használ.

Hogyan nyerjünk?

A táblázat alapján meg tudjuk állapítani az állásokról, hogy azok nyerhetőek-e világossal vagy sem, de ez a táblázat egymagában nem elegendő a játszma megnyeréséhez. Fennáll annak a lehetősége, hogy a táblázat alapján több lépés is nyerő pozícióhoz vezet, de nem tűnik ki, hogy a lépésismétlésre vonatkozó vagy az ötven lépéses szabály nem változtatja-e meg az értékelés eredményét. Ezért újabb szempontokat kell figyelembe venni. A következő három pont elmagyarázása ad arról, hogy több nyerő lépés esetén a program-

nak melyiket kell választania ahhoz, hogy a partit ténylegesen megnyerje:

1. Ha egy nyerő lépés van, akkor azt kell meglépni.

2. Ha a gyaloglépés nyer, akkor lépjen a gyalog!

3. Más esetben olyan lépést kell választani, hogy a király minél feljebb kerüljön a táblán. Ha további választási lehetőség marad, akkor azt kell választani, amelyik a király vonala és a gyalog vonala között a legkisebb távolságot adja. Ha még mindig van választási lehetőség, akkor azt kell választani, amellyel a király a tábla széléhez közelebb kerül.

A bemutatott heurisztikákat nem optimalizáltuk, tehát nem biztos, hogy a program a legkevesebb lépés megtételével fog nyerni, de ha az állás nyerhető, akkor biztosan megtalálja a nyeréshez vezető utat.

A mellékelten bemutatott FORTRAN szubrutin (lásd a *listát*) meghatározza bármely KGK hadállásban, hogy azt megnyerheti-e a gyaloggal rendelkező fél, esetünkben világos. Azokat az állásokat értékeli a szubrutin, amelyekben világos lép. Amikor sötét következik lépésre, akkor egy féllépéssel tovább kell számolni, majd ugyanez az értékelőprogram használható.

KOVÁCS P. ATTILA

ADOK—VESZEK—CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

C16 (64 k) + magnó + joystick + kb 190 program + könyvek (gépi kód, BASIC). Árajánlatot levélben kérek. Fábián Lajos, Sefesd, Széchenyi u., 7562

C16-os és **Plus/4**-es programokat eladok és cserélek. Váczi Rudolf, Kiskunfélegyháza, Klapka u. 25. 6100

Commodore 64-hez GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kbájt. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkezik be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.
Hozott porgramok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-493 este.

Commodore 64-es számítógép tartozékaival eladó! Asztalos Csaba, Balassagyarmat, Rákóczi u. 48. 2660, Tel.: 85-31

C64-re szuper játékprogramokat adok-veszek-cserélek kazettán. Válaszokat listával kérem. Esetleg videokazettáért is adok cserébe programokat. Bánki Péter, Budapest, Közraktár u. 24. 1093

Commodore VC 1541-II floppy vámkezelten, originál csomagolásban eladó. Balla Tibor, Tatabánya VI. Jászai Mari u. 19/3. 2800

Enterprise programok rendelhetők utánvéttel: Disassembler 260 Ft. Copy (másolóprogram) 400 Ft. Egyéb felhasználói programok. Oktatóprogram: Szorzás 220 Ft. Játékprogramok: Kígyó 280 Ft. Tetramino 260 Ft. Hexaletter 220 Ft. Passziánsz 260 Ft. A forgó kocka (demonstráció) 180 Ft. Az öt játékprogram együtt is, kedvezménnyel: 980 Ft. Kívánságra részletes ismertetőt küldök (válaszboritékot kérek). Újlaki László, Budapest, Lavotta u. 24. fsz. 2., 1104

5,25-ös floppy lemez (20 db) eladó. A lemezeknek csak egyik oldala formázható meg. Árajánlatokat a következő címre várok: Boldizsár Gábor, Rém, Petőfi u. 23. 6446. (Csak együtt!)

GEOS program magyar nyelvű fordítása eladó. Vidos Nagy Leona, Budapest, György A. u. 30. A/2. 1125

HOMELAB-3 (64 k) rezidens Debugger és Assembler programokkal, SIO és CTC bővítő panelokkal, alkatrész árért keresi új gazdáját. Nagy Sándor, Pápakovácsi, Fő u. 3. 8596

LACISOFT BASIC Plus/4-re! BASIC kiterjesztés, 15 új utasítás látványos képernyőkezeléshez (nagykarakteres kiírás, scrollok, képcserék). Ára kazettán: 500 Ft. Tape Copy+Nova Turbo (200 Ft). "Novaload" kompatibilis másoló+Turbo (C16-on is egyszerre 12 384 bájt másolható). SZT-BT-TL SOFTWARE CO.; Szabolcsi Tivadar, Kaposvár, Szalma I. köz 5. 7400. Kérjen tájékoztatót!

PC Magazin és **Computer Persönlich** NSZK kiadású folyóiratok 1987-es évfolyamai (52 ill. 26 db) eladók. Bencze József, Kaposvár, 48-as Ifjúság u. 19. 7400

TV-Computer (64 k) sürgősen eladó. Gyári csomagolásban + joysticke + játékprogramok (kazettán), garanciával. Cím: Szolnok, Jászi F. u. 18. X/41. 5000

ZX-Spectrum (80 k), Turbo interfész, kb 200 db program, botkormány eladó. Együtt 18 000 Ft. Szarka Endre, Pápa, Fő u. 24. 8500

ZX-Spectrum (48 k) + Turbo interfész, magnó, joystick, szakkönyvekkel, programokkal 16 500 Ft-ért eladó. Telefon: 732-568 (17 órától)

Sinclair ZX-Spectrumomat sürgősen el szeretném adni. A géphez adok még II. interfészt, joystickot, 2 magyar nyelvű könyvet és programokat. Érdeklődni lehet: Varga Szabolcs Viktor, Budapest, Tóalmás u. 97. 1172, minden nap 17-től.

ZX-Spectrum (48 k), joystick + fényceruza interfész, 160 program, magyar és német nyelvű leírás 15 000 Ft-ért eladó. Morvai Csaba, Budapest, Havanna u. 40. 1181

ZX-Spectrum (48 k) interfésszel, kb. 300 programmal eladó 16 000 Ft-ért. Kotroczó Balázs, Budapest, Hegyalja u. 63. 1124. Tel.: 655-110

ZX-Spectrum, Seikosha GP50S printer, Turbo interfész, ezer programmal és sok szakirodalommal eladó! Érdeklődni: Nagy Szabolcs, Pacsa, Csány László u. 55. 8761

Spectrumhoz, ZX81-hez **TIMEX 2040** nyomtató + 10 tekercs papír 6900 Ft-ért eladó. Vitéz Gyula, Budapest, Zrínyi u. 38. 1039, Tel.: 871-843

Spectrum (teljesen új) programok eladók. Tájékoztatóért küldjön válaszborítékot! Horváth Péter, Siklós, Pf.: 129. 7800

VESZEK

C64-re kitünő programokat keresek. Keresem azokat a DJ-ket akiket érdekel a C64 és a studio-technika egyaránt. Kótai Balázs, Sopron, Frankenburg 8. 9400 -DJM

Vennék használt, de üzemképes C16-os számítógépet magnóval. Az árajánlatokat a következő címre kérem: Ipacs Tamás, Gyomaendrőd, Baross u. 5/1. 5502

Vennék üzemképtelen VC/VIC20, C16, C64, CPlus/4, ZX-Spectrum számítógépeket és tartozékaikat; VC1541, VC1551 lemezegységeket; Quick Shot II. joystickokat; dugaszolható modulokat, valamint Spectrumhoz interfészeket. (Darabok is érdekelnek.) Árajánlatot kérek! Cím: Maróti Gyula, Bordány, Zákányszéki u. 22. 6795

CSERÉLEK

C16, Plus/4 programokat cserélek kazettán. Listát kérek. Sok programom van! Durgonics János, Decs, Tabán 82. 7144

Plus/4 és **C16** programokat cserélek. Batta Zoltán, Dunaújváros, Munkásőr u. 4. II/1. 2400

C64-es programokat cserélek, valamint audio disco zenét adok-veszek-cserélek! Listát a következő címre kérek: Kozák Zoltán, Sopron, Schármár K. u. 2. 9400. Kalandorok kíméljenek!

C64-re mindenféle programot és leírást cserélek. Ezen belül német nyelvtanulást segítő programok érdekelnek. Válaszokat listával várom. Gál Antal, Tiszakécske, Gárdonyi u. 3. 6060

C64-re felhasználói és játékprogramokat cserélek lemezen. Cserealapom kb 500 program. A válaszokat listával kérem: Filep Róbert, Budapest, Szakasits Á. u. 65. IV/33. 1119

Enterprise programot cserélek. Listát kérek! Séra László, Debrecen, Csapó u. 84. 4029

Enterprise 128 programokat cserélek kazettán. Vas Julianna, Jászberény, Messzelátó u. 16. 5100

Sinclair Spectrum (48 k) programokat cserélek. Bankó János, Békéscsaba, Majakovszkij u. 31. 5600

ZX-Spectrum (48 k) programokat cserélek. A válaszokat a programok listájával együtt erre a címre várom: Eötvös Levente, Debrecen, Cserepes u. 22. 4026

Spectrum (48 k) jó állapotban, sok-sok irodalommal és programmal, olcsón eladó. Baranyi Zsolt, Putnok, Rózsadomb u. 8. 3630

ZX-Spectrumra programokat cserélek. Választ listával kérem. Keresem a 3 Weeks in Paradise c. játékot. Csécsei József, Kartal, Vörös H. u. 55. 2173

ZX-Spectrum programgyüjteményemet cserével szeretném bővíteni. Válaszokat listával kérek. Kovács László, Poroszló, Bercsényi u. 1/1. 3388

Ez a rovatunk **KODEX 2000** szövegszerkesztővel készült.

Élvezettel olvasom hónapról hónapra olvasóink leveleit. Megpróbálok úgy válogatni, hogy a dicséreteket is és az elmarasztalásokat is bemutassam. Számomra azért a legtanulságosabbak azok az írások, amelyekben az olvasók a legkülönfélébb kívánságaikat írják meg. Ezekből mutatok be egy kis ízelítőt.

Galacz István, Baja

Atari 800XL-tulajdonos vagyok. A lapjukat több mint egy éve előfizetem. Tekintélyes lapnak tartom a számítástechnikában. „Egy lap olyan, mint az olvasói" — szokták mondani. Sajnos ez a kijelentés és a lap címe is megtévesztő. Csak két típusra szól, ami elég szűk világ. Ráadásul minden újság ezzel a két típussal, a Spectrummal és a C64-gyel foglalkozik.

Egyedül a Budapest Skála Áruház kb. 2600 Atari 800XL-et adott el, de az üzletpolitikából hiányzott a programbehozatal.

Egy NSZK-lap, a Computer Kontakt két példányszámát sikerült megszereznem. Ez a folyóirat nem szerepel a HELIR listáján. Szeretném önöktől megtudni, van-e mód a megrendelésére.

Az is érdekelne, hogy önök közölnek-e a jövőben minden gépre programot, vagy nem érdemes előfizetni a lapra?

Nem pártoljuk se a Commodore-t, se a Spectrumot, se azokat a gépeket, amelyeket vagy egy-egy áruház ügyes üzletkötője talált valamelyik raktárban, vagy élelmes hazánkfiai vettek meg turistaútjukon a kiárusítás alkalmával. Érdemes lenne egyszer felmérni, hányféle gép is van Magyarországon: biztos, hogy háromjegyű számmal lehetne leírni a különféle géptípusokat. Ennyi gépet persze nem tudunk felvállalni. Az átarival kapcsolatban szívesen közlünk írásokat, ha kapunk.

Külföldről az újságokat a Kultúra Külkereskedelmi Vállalat rendeli meg. Azt javaslom, hogy a lapok iránt ott érdeklődjön (Bp. I., Fő u. 30—32. Telefon: 359-370), illetve folyóiratimport-osztály, tőkés reláció (Bp. I., Kuny D. u. 3—5. Telefon: 750-184).

Dankó Péter, Mohács

A januári számban Az olvasó írja rovatban olvastam Murcsó Károly levelét. Ugyan Enterprise-tulajdonos vagyok, mégis van néhány VC—1541-esen írt lemezem, C64-hez írt programokkal. Többek között megvan a G—Pascal és a Max—FORTH. Régen volt már alkalmam C64-et használni, de a G—Pascalhoz van egy demóm. Ez talán nyújtana valami segítséget Murcsó Károlynak.

Írtam Romvári Gábornak (a 47. oldalon „reklámozzák" az Enterprise-klubot, ott szerepel a neve), de azért öntől is tudakozódnék Enterprise-ügyben: milyen floppy és milyen nyomtató illeszthető hozzá? Létezik-e már a PC Mikrovilágban és a µMagazinban is emlegetett irodalom?

A címét elküldtük Murcsó Károlynak. Enterprise-ügyben Romvári Gábor válasza: Minden Shugart 410-es szabványnak megfelelő meghajtó illeszthető az Enterprise-hoz (például a Mitsubishi MF—504), természetesen csak a kontroller kártya segítségével, amit egyelőre a Centrum Áruházak nem árusítanak, de a klubban egy példány megtekinthető. Bármilyen Centronics nyomtató (például a Citizen 120D, a legtöbb EPSON) csatlakoztatható hozzá.

Az irodalom létezik, tudomásom szerint már el is fogyott.

Fülöp Szilárd, Győr

Másfél éve foglalkozom számítógépekkel, és elég korán megéreztem egy programozóklub hiányát városomban. Gondolkoztam, hogyan lehetne egy ilyen klubba bekerülni, de mivel nincs a környéken, egyik programozó ismerősömmel azt találtuk ki, hogy megpróbálunk PC és Commodore szinten programokat írni vállalatoknak: PC-n dBASE III és dBASE III Plus, Commodore-on Oxford Pascal, BASIC és Assembler nyelven. Itt, Győrben már érdeklődtünk vállalatoknál, kedvező eredménnyel.

Nemcsak a levél írójának, de másoknak is üzenem, hogy az NJSZT HCC klub nagyon örülne, ha minél több vidéki klub alakulna. Szándékukat a klub elnökének írják meg, ezt a levelet is neki továbbítottam: dr. Simonyi Endre, NJSZT HCC, Bp. V., Báthori u. 16. Postai cím: 1368 Bp. 5. Pf. 240.

Vékony Sándor, Debrecen

Köszönöm, hogy múltkori levelemet közzétették: így tettem szert három cserepartnerre, és ma már jó hetven program boldog tulajdonosa vagyok. Bár, talán nem is olyan boldog! Hiszen „profi" program, direkt a TVC-re írva, kevés van köztük. A Centrum Áruházak pedig szinte kizárólag az Enterprise-zal foglalkoznak. Kár hazardírozni egy jó számítógéppel, nem figyelni rá egy külföldön már megbukott gépért! Persze az Enterprise nem rossz gép, csak későn indult és lemaradt a szoftvertermésről. Mint a TVC. Csak azt nem értem, ha már van egy jó hazai gyártmányú számítógép, miért kell külföldit behozni? A Centrum Áruházakat ez nem érdekli, legalábbis Debrecenben. TVC-re 4-5 program kapható, ebből is néhányat minden kezdő egy délután alatt megír, a másik gépre viszont 16 programot láttam. Lehet, hogy azok se profik, de az arány elgondolkodtató.

Amit írt, azzal egyetértek, de arra kérem, olvassa el az Enterprise-tulajdonosok korábban közölt leveleit. A küldött programokat átadtam az illetékes szerkesztőnek.

Szabó Sándor, Csopak

Az új év első napjaiban vettem először kézbe a Mikromagazint. Feszült izgalommal lapoztam végig, mint új számítógép-tulajdonos (Atari 600XL), aki minden szakmai ismeretre és tippre éhes. Ez a feszültség nőttön-nőtt, minél jobban belemerültem az írások tanulmányozásába.

Szeretném megtudni, hogyan szerezhetném meg a régebbi számokat, természetesen térítés ellenében. Érdekelne még, hogy az Atari-BASIC kezdőknek és az Atari 800XL c. műveken kívül létezik-e olyan, amelyik bővebb ismeretet ad az atarisoknak, és örömmel olvasnám a gépek szervizelésével és javításával foglalkozó intézmények címeit.

A µM régebbi számait az NJSZT-ben megvásárolhatja (Bp. V., Báthori u. 16. Telefon: 329-349). Ami az Atari-kiadványokat, illetve a szervizzel kapcsolatos információkat illeti: a felsorolt irodalmakon kívül csak angol nyelvű fénymásolt anyag van. Ezzel kapcsolatban az Atari szekciót ajánlatos felkeresni. Az Atari gépek szervizelésével a Novotrade-Fotoelektronik (például az V. kerület, Magyar utcában van egy) foglalkozik.

Tartó Róbert, Leninváros

C64-tulajdonos vagyok. Állandóan változó programkészletem mintegy 1500 különböző célú programot tartalmaz lemezen és kazettán. Soha eszembe nem jutna ezeket a programokat — amelyeket jórészt cserebere útján barátaimtól szereztem — pénzért árulni. Önök miért hajlandók ilyen jellegű hirdetéseket elfogadni és megjelentetni? Nem tilos ez?

Önnek igaza van, mi azonban nem vizsgálhatjuk, hogy a hirdetésben közölt programokhoz a hirdető milyen úton-módon jutott hozzá.

Antal István, Miskolc

Kíváncsian olvastam a különböző számítástechnikai lapokban, hogy ITT A COMMODORE MAGAZIN. Hát erről akarok néhány gondolatot elmondani. 1987. november 7-én feladtam azt az összeget, amely fél évre csillapította volna C-magazin éhségemet. A gubanc: november óta várom a magazinomat, a szerkesztőségtől egy árva levél sem jött, így meglehet, kénytelen vagyok beletörődni, hogy a pénzemet sem látom viszont. Többen kérdezték, hogy jó helyre küldtem-e a pénzt, mire azt válaszoltam, hogy a C-magazin csekkjén fizettem be, hiszen rá volt írva. Hát ezért úgy gondoltam, hogy elküldöm a levelet a Mikromagazinnak és az ÖTLET BIT-LETnek is, hogy aki olvassa, okuljon az én káromon.

Okultunk.

Carlos Baranyai, 273 Pharmacy Ave. #
1112
Scarborough (Ont.) MIL 3E9 Kanada

Múlt évben előfizettem az önök magazinjára. Megkaptam a két első példányt, és nagy érdeklődéssel el is olvastam. 61 éves, Dél-Amerikában született kanadai vagyok. Magyar származású. Bár folyékonyan beszélek magyarul, írni már egy kissé nehéz. Szakmám a fényképezés, szabadidőmben matematikát tanulok és gyakorolom a számítógép használatát. Ha mód van rá, kérem, tegyék lehetővé, hogy kapcsolatba kerüljek magyarokkal, akik szívesen leveleznének a számítógépről. Nekem egy IBM Turbo XT (640 kbájt) gépem van, és sok programom BASIC, Pascal, FORTRAN nyelven, amit szívesen felajánlok bárkinek.

Készséggel közöltük a levelét. Bizonyára számos levelezőpartnere akad az „óhazából".

Balogh Ádám, Kiskunhalas

1987 nyara óta van egy Enterprise 128-as számítógépem, azon kezdtem el programozni. Könyvből tanultam lelkesen, így állíthatom, hogy legalább kétharmadát ismerem és használom az Enterprise BASIC-jének. A Mikromagazinnak csak az idei első számát láttam, de teljesen magával ragadott. Annyi érdekes dolog van benne, hogy a bennem felgyülemlett gondolatoknak csak egy részét tudom leírni.

Első problémám, hogy mi az az NJSZT HCC? Én szívesen társulnék a többi Enterprise-tulajdonossal, de hogyan, ha nem tudom, mi az, amit ehhez meg kell keresnem.

Felfigyeltem Göbölös László kérésére: ha tudnak Enterprise-tulajdonosról, írják meg a címét! Ha egy vidéki is jó, hát a címem: Balogh Ádám, Kiskunhalas, Bercsényi u. 2. 6400. Ugyanez az olvasótárs kritizálta a géphez kapott felhasználói kézikönyvet. Nos, ebben nem egyezik a véleményünk. Szerintem egyedülállóan szuper. Viszont tudom, hogy létezik technikai ismertető, ugyanis nekem is van — ha érdekli.

Krasznai Éva cikkére válaszolva: én ugyan nem vagyok Kovács Béla Pusztaszabolcsról, de én is örömmel vállalnám az Enterprise gépekre érkező programok ellenőrzését.

A szerkesztőséget kérném, ha tudnak Enterprise másolóprogramról, azt küldjék el vagy tegyék közzé!

Ha érdekli a szerkesztőséget, elküldhetem „zongora"-programomat. Alapszintű hangkeltésre dolgoztam ki, s szándékomban áll később továbbfejleszteni. Végezetül kérném, hogy ha tudnak valakit, aki cserélne Enterprise-programokat, írjanak!

Végre egy boldog Enterprise-tulajdonos!

Csak ismételni tudom: várjuk olvasóink programjait, írásait, leveleit. Egy lap értékét az jelzi, hogy az olvasók milyen mértékben tartják a kapcsolatot a szerkesztőséggel. Mi egyelőre még nem panaszkodhatunk. Ezt reméli továbbra is

KOVÁCS GYŐZŐ

A TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.

Horváth Zsuzsa 665-011/2663 mellék
vagy 813-197

Budapest, Pf.: 454, 1372

A két Zátonyi — a fizikáról C16-on és Plus/4-en

A cím ugye egy szindarabra emlékeztet, de azért nem kell mindjárt valami nagy drámát sejteni mögötte. Egyszerűen arról van szó, hogy apa és fia együtt vágtak neki az általános iskolai fizika számítógépesítésének. Programjaik még elég gyorsan elkészültek, de a könyv, amit mellékelni szerettek volna, az bizony Luca széke lett, bár nem az ő hibájukból. Egy kiadóra persze nem éppen szégyen, hogy valamit annyi idő alatt hoz napvilágra, mint amennyi ideig egy Luca székét készíteni illik, csakhogy esetünkben a Luca-naptár több, mint egy igazi évet tett ki. A „két Zátonyi" mindenesetre joggal mérges a dolog elhúzódása miatt. Különösen, mert — mint azt szubjektív véleményként előre jelezhetjük — nagyon hasznosat alkottak.

Az új tesztalanyra a Novotrade Deltasoft irodájának új termésében böngészve bukkantunk. Közben töredelmes vallomásokat hallgattunk a fenti malőrről. Mi persze inkább programcsomagot akartunk bírálni és nem készítési körülményeket. Örülünk, hogy van végre egy jó általános iskolai fizikaoktató programcsomag, amelyet Plus/4-en vizsgáltunk meg. A termék összefoglaló adatait táblázatban közöljük.

ÖSSZEFOGLALÓ ADATOK	
Forgalmazó:	Novotrade Deltasoft
Terméknév:	Fizika oktatása C16, Plus/4 számítógéppel
Szerző:	dr. Zátonyi Sándor, Zátonyi Sándor
Géptípus:	C16, Plus/4
Hordozó:	kazetta vagy floppy
Dokumentáció:	C16, Plus/4 az általános iskolai fizikatanításban (238 oldal!)
Ár:	kazettán: 376 Ft floppyn: 600 Ft

Nos, az ár floppy-változat esetén nem egy tanuló zsebpénzéhez méretezett, de ezért a pénzért a vásárló kap egy jó könyvet is, amelyben a hasznos tantárgy-pedagógiai információkon kívül benne van az összes program listája. A programok ezért továbbfejleszthetők, amire a szerzők — helyesen — fel is szólítanak. A pedagógus a listák birtokában a saját tanítási szokásainak megfelelően tudja átgyúrni a programokat. Persze el is ronthatja azokat, de helyes, ha emiatt nem vonjuk el tőle a programok módosítási lehetőségét. Itt nincs olyan éles garanciakérdés, mint az üzleti szoftverben.

A programcsomag 46 (45 tantárgyi és egy betűgrafikai) programból áll. Nem használja ki a magyar ékezetes betűkészletet, mert az a tantárgy szempontjából nem lényeges. Kihasználja viszont a botkormány-csatlakozást mérések céljára, ami szemléletesen mutatja, hogy nemcsak szoftverrel érdemes foglalkozni egy számítógép használatakor. A billentyűzet, mint egyedüli adatbeviteli eszköz, lassú, és bizonyos adatok bevitelére eleve alkalmatlan. A számítógépek külvilággal létesítendő kapcsolatának oktatása ma is túl szűk körű, de még nem vált elég nyilvánvalóvá, hogy a jövő, a fejlődés szempontjából nagy hiba csak a programozás oktatását erőltetni. Különösen, mert ma már hazánkban is elmondható, hogy a csapból is szoftver folyik. Ennek ellenére óriási tömegben képezünk ki programozókat. Nem bánjuk majd meg?

A programcsomag négy fő fizikai szakterületet érint: mechanika, hőtan, fénytan, elektromosság. A programok közül néhány egy-egy témakör köré csoportosul (sűrűség, sebesség, munka, hatásfok, forgatónyomaték, ellenállás, melegedés-hűlés), míg a többi egyedi témájú. A programok egy része gyakoroltatja a témát és méri a tanuló teljesítményét, mások pedig szemléltetik a fizikai jelenségeket. Nagyon fontos csoportot alkotnak a fizikai mennyiségek nagyságrendi változásait táblázatokkal bemutató programok. Jellegzetes probléma — még az egyetemi hallgatóknál is, nemhogy az általános iskolában! —, hogy a tanulók nem érzékelik a nagyságrendi hibákat. Fontos tehát, hogy táblázatok formájában lássák, mi történik, ha egyik vagy másik mennyiséget vagy paramétert változtatjuk.

A gyakoroltató programok szervezése ügyes. Egy hibázást megengednek, de pontot vonnak le érte. A megoldást is megadják a HELP gomb megnyomására, ám ezeket a kérdéseket kihagyják az értékelésből: nem adnak rá pontot, de nem is számítják be a feladott kérdések számába, ami vitatható. A gyakorlatok a tanulót fejszámolásra kényszerítik. Ez nem helytelen, de az ügyeskedők hamar ki tudják kerülni ezt az akadályt. A lehetséges megoldások kitalálását az olvasóra bízzuk.

Jó ötlet a kissé kevés szövegváltozattal konstruált, de érdekes, random adatokkal feladott szöveges példák szerepeltetése. A tanulók valamiért mindig jobban félnek a szöveges példáktól, pedig az életben rendre azzal találkoznak. Nem árt, ha próbáljuk őket időben megbarátkoztatni velük.

Az említett táblázatokat bemutató programoknál lehetett volna még szemléletesebbé tenni a változásokat, ha a + és −gombokkal változtathatnánk az egyes oszlopok mennyiségeit. Azért persze így is izgalmas, amint a program egy oszlop megváltoztatásakor gyorsan átszámolja a többi oszlop adatát (mini-VisiCalc).

A grafikonokat rajzoló programok különösen szépek, bár programozásuk nem túl bonyolult. Ez fontos is, ha nem akarjuk azt, hogy a programot módosítónak túl sokat kelljen bajlódnia a körítéssel, ahelyett, hogy a problémára tudna koncentrálni. Jó ötlet, hogy a telefirkált ábrákat is le lehet tisztítani.

A programcsomag túl nagy ahhoz, hogy a részletekre kitérjünk az elemzésünk keretében. Azért elmondjuk, hogy melyik programok tetszenek nekünk a legjobban: szabadesésmérés, színek, tükrök-lencsék, Ohm törvénye, rezgőmozgás, kapcsolások, V—A műszer.

A programok ügyesen használják ki a számítógép adottságait. A grafikus lehetőségek persze nagyon is fontosak a mondanivaló alátámasztására. A felbontás gyengéi csak néha zavaróak. Alapos munkát takar a színek olyan összeválogatása is, hogy a kép mind színesben, mind fekete-fehérben jól látható. A színeket bemutató program persze színes megjelenítőt igényel, de erre a normál tévé jobb, mint az igazi monitorok, mert a színkeverés inkább rossz képalkotásnál látszik jól, vagy távolabbról kell nézni a monitort.

A vezérlőbillentyűkre jobb lett volna emlékeztetni az alsó sorban, mert azok szerepe csak a bevezetőben jelenik meg, amit az ember hamar elfelejt, és esetleg a könyv sincs kéznél. Sajnos a HELP gomb hatása olykor katasztrofális lehet: a program egyes esetekben teljesen abbahagyja azt a funkciót, amelyben vagyunk. Ez a BASIC megvalósítás miatt persze érthető, de a könyvbeli hivatkozás alapján nem ezt várná az ember. A könyvben nem szerepel, de grafikus képeknél a lelövés (RUN/STOP) előtt célszerű kivárni, amíg a HELP gombra reagál a program. A HELP gombbal ugyanis ismét TEXT módba (GRAPHIC 0) vált, és nem kell erőlködni a GRAPHIC 0 „vak" beírásával, hogy láthatóvá tegyük a szövegeinket.

A programcsomagot majdnem végig kiválónak minősítettük, mint az a minősítő táblázatból látható.

MINŐSÍTŐ ADATOK	
Kezelhetőség:	jó
Teljesség:	kiváló
Dokumentáltság:	kiváló
Használhatóság:	kiváló
Ár/teljesítmény:	kiváló
Összbenyomás:	kiváló

A programcsomagban jó programozástechnológiai eszközök is vannak, amelyek a forrásnyelvű listáknak köszönhetően közkinccsé válhatnak. Bár a teljességre kiváló minősítést adtunk a programcsomagban érezhető, nagy erőfeszítést igénylő munkára tekintettel, még nagyon sok ilyenre lesz szükség ahhoz, hogy az egész fizikát jól adaptáljuk. A szerzők — mint már említettük — fel is szólítanak a továbbfejlesztésre.

A programcsomag készítőinek ezúton is gratulálunk, és reméljük, hogy most már tényleg gyorsan és széles körben eljutnak programjaik a tanárokhoz és a nebulókhoz. Ugyanakkor a fizikatanároknak, szakköri tagoknak is sok sikert kívánunk a programcsomagban megtestesülő nagy tudásanyag asszimilálásához. Megjegyzendő, hogy a programcsomag önálló tanulás támogatására is alkalmas, és nemcsak az általános iskolások okulhatnak belőle.

ZSADÁNYI PÁL — ifj. ZSADÁNYI PÁL

Szűcs Pál:
Számítógépes oktatási programok tervezésének módszertana (Budapest, 1987. OMIKK, 306 oldal. Ára: 120,— Ft.)

A számítógépek oktatásbeli alkalmazásának legfontosabb vonása az, hogy a tanítási-tanulási folyamatban új módszerek megjelenését teszik lehetővé, miközben az egyes tanulók igényeihez alkalmazkodnak. Terjedését és gyakorlati felhasználását jelentősen gátolja az a tény, hogy világszerte különböző, egymással nem kompatibilis berendezések állnak csak rendelkezésre. Jelenleg nincs érvényben megfelelő szabványgyűjtemény, ajánlás a hardverrel és szoftverrel kapcsolatban.

A tanulók és tanárok számára ezért ma nem az a fő feladat, hogy a divatos, mesterséges gépi nyelveket sajátítsák el és programokat írjanak, hanem az, hogy a számítógépes feladatmegoldások logikáját megismerjék és a tervezés módszertanát alkalmazzák a munkájukban.

A könyv célja, hogy a tanároknak és a tanulóknak ezeket az ismereteket megfogalmazza, rendszerezze és példákkal tegye még érthetőbbé. Segítséget nyújt mindazoknak, akik a korszerű oktatási folyamatban a hatékonyságnövelés és eredményesség egyik fontos eszközének tekintik a személyi számítógépet.

Szilasi Anna:
Mikroszámítógépes informatikai rendszerek és hálózatok az egészségügyben (Budapest, 1987. LSI ATSZ, 357 oldal.)

Az állampolgárok megfelelő egészségügyi ellátásának egyik alapvető feltétele, hogy az egészségügyi tevékenységet végző szervek, intézmények, az egyes orvosok korrekt, pontos és naprakész információkkal rendelkezzenek a hozzájuk forduló személy korábbi megbetegedéseiről, vizsgálati eredményeiről, orvosi kezeléséről.

Az információk hiánya az azok pótlására fordított idő és költség miatt lassítja és megdrágítja az egészségügyi ellátást, emellett olyan kapacitásokat köt le, amelyek egyébként is viszonylag szűkösek.

A jelenlegi helyzet, a reális szükségletek és a gazdasági érdekek egyaránt sürgetik az egészségügy információs rendszerének érdemi fejlesztését, az információkezelés módszereinek és eszközhátterének alapvető javítását.

Ezt felismerve döntött Pécs Város Tanácsa egészségügyi osztálya egy, a városi egészségügyi hálózat információit kezelő számítógépes egészségügyi informatikai rendszer kifejlesztéséről.

A kiadvány célja az egészségügyi informatika elterjesztésének elősegítése: a kritikus pontok bemutatása, az egészségügyi mikrogépes központok megtervezésének, működtetésének elősegítése. Tartalmazza a jelenlegi helyzet felmérését, elemzését, a javasolt rendszer logikai modelljét, megvalósításának, bevezetésének és üzemeltetésének módját.

Knuth, Donald E.:
A számítógép-programozás művészete. 1. Alapvető algoritmusok (Budapest, 1987. Műszaki Könyvkiadó, 654 oldal. Ára: 180,— Ft.)

Knuth, Donald E.:
A számítógép-programozás művészete. 2. Szeminumerikus algoritmusok (Budapest, 1987. Műszaki Könyvkiadó, 690 oldal. Ára: 180,— Ft.)

A szerző könyvsorozatának előszavában így vall a programozásról: „Számítógépprogramokat készíteni különleges, vonzó feladat: nemcsak gazdasági vagy tudományos hasznuk miatt, hanem mert — mint a versírás és zeneszerzés — esztétikai élményt is nyújtanak. Ennek a hét kötetre tervezett könyvsorozatnak az a célja, hogy a programozói mesterség különböző fogásaival és részleteive! ismertesse meg az Olvasót.

Nem a számítógép-programozás bevezetőjének szánjuk a következő fejezeteket; feltételezzük, hogy az Olvasó már rendelkezik bizonyos jártassággal ezen a téren. Lényegében igen kevés előismeretet tételezünk fel: a kezdőnek azonban időre és gyakorlásra lesz szüksége, ha világos képet akar nyerni arról, mi is az a digitális számítógép.

Az Olvasóról feltételezzük, hogy

a) van némi fogalma egy tárolt programú számítógép működéséről; . . .

b) képes a problémák megoldását olyan explicit módon megfogalmazni, hogy azt a gép »megértse«; . . .

c) ismeri valamennyire a legelemibb számítógépes módszereket; . . .

d) nem idegenek előtte a legalapvetőbb szakkifejezések, mint pl. tár, regiszterek, bit . . .

Ez a négy feltétel talán összefoglalható egyetlen kritériumban, amely így szól: az Olvasó már megírt és hibátlanul lefuttatott mondjuk legalább négy programot, legalább egy számítógépen.

Ezek a könyvek elsősorban kézikönyvek, amelyek több fontos témakör eddig elért eredményeit foglalják össze. Emellett tankönyvként is használhatók, egyéni tanulásban, vagy főiskolai, egyetemi szemináriumokon a számítógépés informatikai tudományokban.

A könyvsorozatot olyanok számára írtam, akik nem pusztán alkalmi érdeklődéssel foglalkoznak számítógéppel. Azonban semmiképpen sem csak számítógépes szakemberekre gondoltam, sőt egyik legfontosabb célkitűzésem éppen az volt, hogy hozzáférhetővé tegyem ezeket a programozási módszereket más területen dolgozók számára, akik jó hasznát vehetik a számítógépnek."

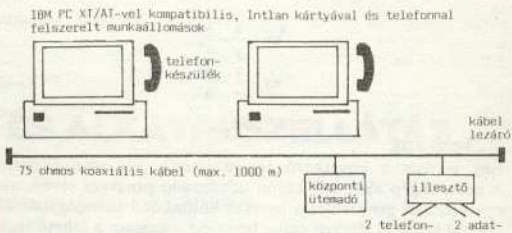
Intlan

A Budapesti Műszaki Egyetemen kifejlesztettek az IBM PC-vel kompatibilis mikrogépekre egy olyan helyi hálózatot, amely lehetővé teszi, hogy bármely két munkaállomás között ne csak adatforgalom, hanem hangforgalom is megvalósuljon. Ez azt jelenti, hogy az adatok átküldése mellett át is lehet szólni, hogy például „Mancika, tegye félre az előző ügyet; amit most küldök, az a legsürgősebb!"

Fizikailag a rendszer max. 20 db, IBM PC-vel kompatibilis mikrogépből áll, melyek egy max. 1000 m hosszú, mindkét végén lezárt hajlékony koaxiális kábelre vannak felfűzve. A gépekbe bele kell helyezni egy-egy Intlan kártyát, mondjuk a Pcnet vagy az Arcnet kártya helyett. Ezután a Novel NetWare szoftvernek a Budapesten kifejlesztett Intlant alkalmazó verzióját kell betölteni. Ezután mind az adatátvitelt, mind a hangátvitelt az Intlan kártya vezérli. A hangátvitel bármely két állomás között megvalósítható, függetlenül attól, hogy az adott két állomás között az adott pillanatban folyik-e adatátvitel.

Az Intlan hálózat csatlakoztatható nyilvános telefonhoz, továbbá lehetőség van több Intlan hálózat összekapcsolására is.

Az Intlan-hálózat már üzemel a XX. Kerületi Tanács ügyfélszolgálati irodájában.



Az Intlan-hálózat sémája

Blicc-stop

A fenti elnevezésű elektronikus adatrögzítő készüléket a Műszertechnika Kisszövetkezet a Kisalföld Volánnal közösen fejlesztette ki. A készülék, amely az autóbuszok műszerfalán helyezkedik el, egyrészt utasszámlálást végez, másrészt kalauzként is működtethető. Ez azt jelenti, hogy a jegykezelő és az ellenőr feladatát is ellátja.

A készülék kijelzőjén minden megállónál megjelenik az ajtókon fel- és leszálló, illetőleg a járművön lévő utasok száma. Ezeket az adatokat cserélhető félvezetős memória rögzíti, ami ezenkívül azt is „megjegyzi", hogy az autóbusz mennyi időt töltött a megállókban, milyen sebességgel és mennyi idő alatt tette meg az utat az egyik megállótól a másikig. Az adatok összesítése hozzásegíti a közlekedési vállalatokat ahhoz, hogy a járatok menetrendjét gazdaságosan, a tényleges igények figyelembevételével alakítsák ki.

Amint a mikrogép érzékelőit a vonaljegy-érvényesítő készülékekbe is beépítik, össze tudja vetni az érvényesített jegyek számát az utasok létszámával. A fejlesztők gondoltak a bérletesek ellenőrzésére is: a bérleteket egy erre a célra kialakított berendezés kontrollálja.

Pontvadászat

Új rejtvényt indítottunk második számunkban útjára, remélve, hogy elnyeri olvasóink tetszését. Minden számunkban két feladatot közöltünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után közöljük az elérhető maximális pontszámot. A részmegoldásokat is pontozzuk.

A pontgyűjtést, vagyis a pontvadászatot az esztendő végén zárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvutalványt is kapnak.

A helyes megoldások a feladatok közreadása után két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. július 15.

Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433.

Jó vadászatot kíván a feladatok összeállítója:

dr. Hoffmann Tibor

A 88/5. szám 1. feladatába sajnálatos hiba csúszott be.

A feladat 3. sorában „. . . a másik oldalt érinti" helyett „. . . a másik két oldalt érinti" szöveg a helyes.

1. feladat

Oldjuk meg a következő egyenletrendszert:

$$xy-\frac{x}{y}=\frac{3}{4},$$
$$xy-\frac{y}{x}=\frac{4}{3}.$$

(4 pont)

2. feladat

Egy játékos a következőképpen játszik egy számítógéppel. A számítógép akár egy külön adatbeadó program révén, akár egy véletlen generátorral nyert *5* különböző számjegyből álló számot tárol. A játékos célja, hogy ezt a számot a lehető legkevesebb próbálgatással találja ki. Egy *ötjegyű* próbaszám beadása esetén a számítógép egy kétjegyű számmal válaszol, melynek az első jegye azt adja meg, hogy hány számjegyet talált el a próba, a második pedig, hogy ezek közül hány számjegy van a helyén. Nyilván ez a válasz *0*-tól (egyetlen szám sem egyezik) *55*-ig (a szám kitalálása) változik úgy, hogy a második számjegy soha nem lehet nagyobb az elsőnél.

Írjunk szubrutint arra, hogy *5* egymás után beadott számjegyből a próbaszámot képezze a gép. (3 pont)

Írjunk szubrutint arra, hogy egy *ötjegyű* számból az *5* különböző jegyet határozza meg a gép. (4 pont)

Írjunk szubrutint arra, hogy adott tárolt szám mellett a játékos által beadott vagy valamilyen módon beadott számra a fent vázolt választ adja meg a gép. (8 pont)

A szubrutinok bármilyen gépre megírhatók. Egy későbbi feladványban ezeket fel fogjuk használni.

(A feladat egy részét Nagy D. István csíkszeredai olvasónk küldte be Romániából.)

Az 1988/4. szám feladatainak megoldása

1. feladat

Mint az 1988/3. szám 1. feladatának megfejtéséből látható, a tétel igaz az első, a második és a harmadik hatványokra.

Tételezzük fel, hogy az első k hatványra a tétel igaz. Ekkor a $k+1$-edik hatványt a következőképpen írhatjuk fel:

$$i^{k+1}=\frac{(i+1)^{k+2}-i^{k+2}-A_1i^k-A_2i^{k-1}\ldots-1}{k+2},$$

ahol $A_1, A_2, \ldots$ k-tól független számok.

Az első k hatványra igaz lévén a tétel, a $k+1$-edik hatványok összegét így írhatjuk fel:

$$\sum_{i=1}^{n} i^{k+1}=\frac{(n+1)^{k+2}-1-C_n(n+1)-n}{k+2}=$$
$$=\frac{n+1}{k+2}[(n+1)^{k+1}-C_n-1]=D\frac{n(n+1)}{k+2},$$

ahol C és D n-től is függő állandó, és az előző formulában felhasználtuk azt, hogy a szögletes zárójelben minden tag osztható n-nel, kivéve az $n+1$ hatványozás elvégzésénél az 1 és a végén elhelyezkedő -1, melyek viszont 0 összeget adnak. (5 pont)

2. feladat

Legyen a szubrutin hívásának a száma a programban h. Ez annyit jelent, hogy szubrutin nélkül ennyi helyen fordul elő egyenként m memóriahellyel a tevékenység, vagyis a programban hm helyet foglal el. Ha a szubrutint alkalmazzuk, akkor a memóriaigény

$$h(5p+8)+m+2p+4$$

Így tehát h-t úgy kell meghatároznunk, hogy

$$h(5p+8)+m+2p+4<hm$$

legyen.

Ez h-ra azt jelenti, hogy

$$h>\frac{m+2p+4}{m-5p-8}.$$

Ebből az is látszik, hogy az adott gépen csak olyan szubrutint érdemes alkalmazni, melynek magjára fennáll

$$m>5p+8.$$

Ennél kisebb mag esetében értelmetlen a szubrutin használata. (Persze ez csak a memóriahely foglalása szempontjából lényeges, mert lehet, hogy a program áttekinthetősége mégis megéri a szubrutin használatát.) (4 pont)

Paraméter nélküli szubrutinnál a fentiek szerint

$$h>\frac{m+4}{m-8},$$

ami azt jelenti, hogy például egy $m=12$ magú szubrutin esetében csak $h>4$ esetében gazdaságos a használat. Ha a mag kisebb, akkor még magasabbra emelkedik a gazdaságossági határ. Például $m=9$ esetében csak $h>13$ esetén érdemes a szubrutint alkalmazni.

Ha 1 paraméteres a szubrutin, akkor

$$h>\frac{m+6}{m-13}$$

Ebben az esetben tehát csak 13 memóriahelynél nagyobb magot érdemes szubrutinba venni és $m=14$-nél csak 20 használatnál több esetben, $m=32$-nél pedig még mindig legalább 3 alkalmazásnak kell lennie, hogy gazdaságos legyen.

Ha 2 paraméteres a szubrutin, akkor

$$h>\frac{m+8}{m-18}.$$

Itt tehát $m>18$-nál lehet csak gazdaságos a szubrutin használata és $m=19$-nél minimálisan 27-nél több hívás esetén. (1 pont)

Felvethetjük a kérdést úgy is, hogy mekkora magnagyság esetén lesz már 2-nél több alkalmazásnál a szubrutin gazdaságos. Erre azt kapjuk, hogy

$$m>12p+20$$

esetén.

Az ECONORG BUDAPEST a Multitech

mai nevén ACER gépcsalád teljes választékát kínálja

Teljes termékskála a 16 és 32 bites mikroszámítógépek körében

A 32 bites gépcsaládból ajánljuk:

ACER SYS–32/20
Intel 80386, 20 MHz, 2–16 MB memória, 3–35 terminál, 50–380 MB winchester, SCO Xenix System V.2 operációs rendszer

ACER 1100
Intel 80386, 4,77–16 MHz, 1–16 MB memória, 40–140 MB winchester, MS-DOS operációs rendszer

A 16 bites gépcsalád tagjai

ACER 910
Intel 80286, 8–12 MHz, 1–16 MB memória, 40–80 MB winchester, MS-DOS

ACER 710
Intel 8088–1, 4,77–10 MHz, 256–768 KB memória, 20–40 MB winchester, MS-DOS

ACER 500+
V20–8, 4,77–8 MHz, 256–640 KB memória, 20 MB winchester, MS-DOS

TELJESKÖRŰ PERIFÉRIA- ÉS ALKATRÉSZELLÁTÁS

Monochrom és színes monitorok, normál és EGA kivitelben
20, 40, 53, 86 Mbyte kapacitású winchesterek
240, 300 karakter/másodperc sebességű mátrixnyomtatók
Lézer nyomtatók, Scanner-ek, Streamer-ek, Mous-ok

Teljes körű kiegészítő és bővítő kártyaválaszték:

Memóriabővítések
Matematikai koprocesszorok
Multifunkciós kártyák
Grafikus kártyák

Lokális hálózatok:

ComcoLan lokális hálózat 1–255 munkahelyig:
Hálózati kártya
Hálózati operációs rendszer
Passzív és aktív HUB-ok
Kábelezés

100%-osan LICENC-TISZTA, IBM kompatibilis termékcsalád:
16 és 32 bites mikrogépek komplex, fokozatosan bővíthető architektúrában

Távadatfeldolgozás:

ComcoMODEM (1200–2400 BPS), Stand-alone modemek, Add-on kártya kivitelű modemek, ComcoTelex

Vegye igénybe az ECONORG komplex mikrogépes szolgáltatását:

– helyzetfelmérés,
– rendszertervezés,
– programozás,
– bevezetés,
– oktatás–tanácsadás,
– rendszerkövetés,
– szerviz (garanciális és garancia utáni),
– magyar nyelvű dokumentációk

ajánlja saját fejlesztésű alkalmazói programcsomagjait, felhasználói rutinjait.

Bővebb információ:

ACER

Központ: Budapest, XIV., Ajtósi Dürer sor 10.
Postai cím: 1393 Budapest, Pf. 319.
Telefon: 421-974 Telex: 22-6544
Budapesti szaküzlet:
Budapest, VI., Szinyei-Merse u. 1.
Telefon: 127-628 Telex: 22-6684

Győri kirendeltség:
Győr, Lukács Sándor u. 17.
Telefon: 96-14808 Telex: 02-4679
Salgótarjáni kirendeltség:
Salgótarján, Ady E. u. 1.
Telefon: 32-10971 Telex: 22-9380

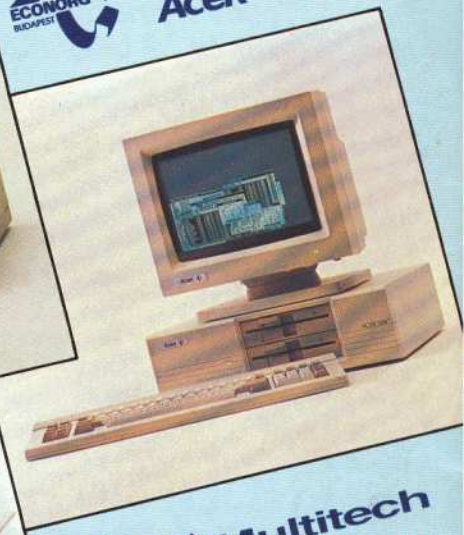
Keresse fel önálló pavilonunkat a BNV-n Személyesen győződjön meg szolgáltatásaink színvonaláról

ECONORG BUDAPEST

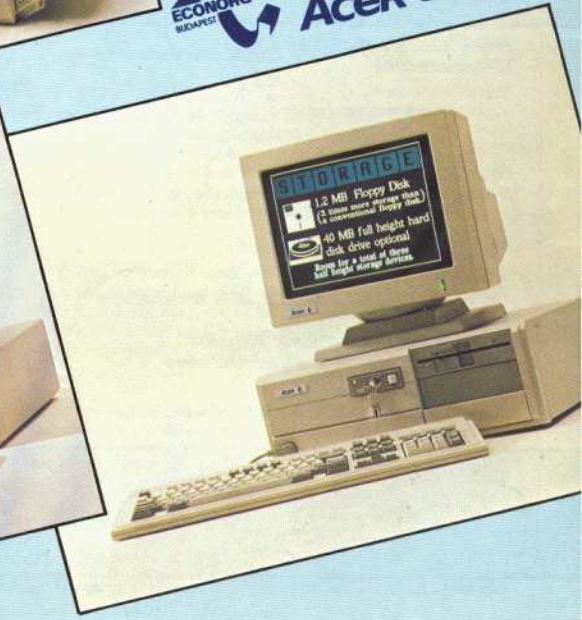
Multitech

Acer 1100

Acer 500



Acer LP-75

Acer 710

Acer 910